

# PATENT ABSTRACTS OF JAPAN

(11) Publication number : 2000-259429  
 (43) Date of publication of application : 22.09.2000

(51) Int.CI.

**G06F 9/46**  
**G04G 15/00**

(21) Application number : 11-065502  
 (22) Date of filing : 11.03.1999

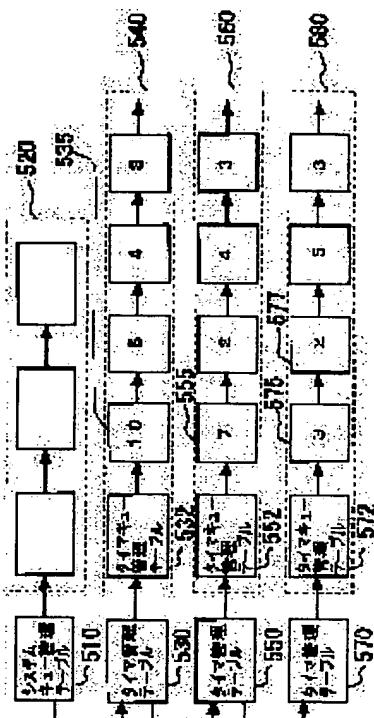
(71) Applicant : MITSUBISHI ELECTRIC CORP  
 (72) Inventor : KAMOKI ATSUSHI

## (54) DEVICE AND METHOD FOR MANAGING TIMER

### (57) Abstract:

**PROBLEM TO BE SOLVED:** To operate plural various systems with one timer device by unitarily managing the timer events of plural systems.

**SOLUTION:** Timer queue elements 535 or the like of one (sub) system are respectively connected in the shape of queue and managed for each (sub) system by a timer queue managing table 532 or the like. The (sub) system is managed by a timer managing table 530 or the like for each (sub) system and each (sub) system is managed by a system queue managing table 510. An idle timer queue 520 successively connects used timer queues. In the case of connecting a novel timer queue element to a timer queue element 540 or the like, when there is the timer queue element 535 or the like, that novel timer queue element is connected by calculating a differential timer value but when there is no timer queue element, that novel timer queue element is connected to the head of the timer queue 540 or the like. Thus, plural timer queues are integrated, timer events are unitarily managed and plural various (sub) systems can be operated with one timer device.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

\* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

CLAIMS

---

[Claim(s)]

[Claim 1] Timer management equipment which is characterized by providing the following and which manages a hawksbill turtle vent by the queue. These timer management equipment is one or more timer queue means to hold the timer queue element corresponding to a hawksbill turtle vent by the queue according to the publishing agency of a hawksbill turtle vent, and this timer queue element has the difference of the deadline time of a hawksbill turtle vent, and the time connected to this timer queue as a timer value. A selection means to choose the minimum timer queue element which has the minimum timer value in the timer value which the separate timer queue element held at the head of the separate timer queue means about the one or more aforementioned timer queue means has when timer interruption occurs. A rearrangement means to hold at the head the timer queue element held at the degree of this minimum timer queue element, and to delete this minimum timer queue element from this timer queue means about a timer queue means to hold the aforementioned minimum timer queue element. A reconfiguration means to newly reconfigure the value which subtracted the minimum aforementioned timer value from the timer value which each timer queue element held at the head of each timer queue means has about other different timer queue means from a timer queue means by which the aforementioned minimum timer queue element was held as a timer value.

[Claim 2] The separate timer queue managed table means respectively corresponding to the timer queue means of the aforementioned exception individual, It has further a system queue managed table means to manage the timer queue managed table means of the aforementioned exception individual in predetermined order. the aforementioned selection means Timer management equipment according to claim 1 which chooses this aforementioned timer queue managed table means separate in predetermined order, and is characterized by choosing the minimum timer queue element about the separate aforementioned timer queue means corresponding to the separate aforementioned selected timer queue managed table means.

[Claim 3] The aforementioned system queue managed table means is timer management equipment according to claim 2 characterized by managing independently other timer queue managed tables on which the aforementioned timer queue managed tables differ when the operating state of the issue origin of the hawksbill turtle vent corresponding to the aforementioned timer queue managed table changes.

[Claim 4] The aforementioned system queue managed table means is timer management equipment according to claim 2 characterized by managing independently other timer queue managed tables on which the aforementioned timer queue managed tables differ when an issue origin's of hawksbill turtle vent corresponding to aforementioned timer queue managed table existence state changes.

[Claim 5] The aforementioned rearrangement means is timer management equipment according to claim 1 characterized by to re-connect the deleted aforementioned minimum timer queue element to the position corresponding to this predetermined periodic time [ in the aforementioned timer queue means ] after when the aforementioned minimum timer queue element deleted from the aforementioned timer queue means corresponds to the periodic hawksbill turtle vent started for every predetermined periodic time.

[Claim 6] It has further the connecting means which newly connect the timer queue element corresponding to a hawksbill turtle vent to the aforementioned timer queue means. these connecting means In the state where the 1st timer queue element corresponding to the periodic hawksbill turtle vent started for every 1st period time is already connected to the aforementioned timer queue means When the 2nd timer queue element corresponding to the periodic hawksbill turtle vent started for every 2nd period time which has the 1st period time and a predetermined relation is newly connected to the aforementioned timer queue means, Connection which makes this time earliest deadline time of the 1st timer queue element and 1st deadline time of the 2nd timer queue element is made. 1 time of timer interruption -- the [ the 1st timer queue element and ] -- the timer management equipment according to claim 1 characterized by increasing the number of times which makes the deadline of 2 timer queue element pass simultaneously, and cutting down the consumed electric current by timer interruption

[Claim 7] The aforementioned predetermined relation with the 1st period time which the 2nd period time has is a relation whose 2nd period time is the multiple of the 1st period time. the aforementioned connecting means The value which lengthened the present time from the earliest time whose deadline the 1st timer queue element passes is connected as 1st timer value of the 2nd timer queue element. Timer management equipment according to claim 6 characterized by making deadline time of the 1st timer queue element, and deadline time of the 2nd timer queue element into this time.

[Claim 8] It is timer management equipment according to claim 6 characterized by to connect the 2nd timer queue element at the earliest time when the aforementioned predetermined relation with the 1st period time which the 2nd period time has is a relation whose 2nd period time is the divisor of the 1st period time at, and the time which lengthened the integral multiple of the 2nd period time from the earliest time when the 1st timer queue element passes the deadline of the aforementioned connecting means does not become later than the present time.

[Claim 9] For the aforementioned connecting means, the 2nd period time is timer management equipment according to claim 6 characterized by being a relation with the aforementioned as relatively prime predetermined relation with the 1st period time which the 2nd period time has as the 1st period time, and connecting so that the 1st timer queue element and the 2nd timer queue element may pass the deadline of simultaneously for every period time used as the least common multiple of the 1st period time and the 2nd period time.

[Claim 10] It is timer management equipment according to claim 1 characterized by to have further the connecting means which newly connect to the aforementioned timer queue means the timer queue element corresponding to the hawksbill turtle vent which has predetermined warm-up time, and for these connecting means to have the number of times repeatedly with the periodic time based on the aforementioned predetermined warm-up time for the aforementioned timer queue element, and to change into the periodic timer queue element corresponding to this periodic hawksbill turtle vent started by the number of times of a repeat for every periodic time of this, and to connect.

[Claim 11] The aforementioned timer queue element is further equipped with the allowable error over the aforementioned timer value. the aforementioned rearrangement means When the aforementioned allowable error which the timer queue element held at the degree of this minimum timer queue element has about a timer queue means to hold the aforementioned minimum timer queue element is smaller than the

value which subtracted the minimum aforementioned timer value from the aforementioned timer value, this -- the timer management equipment according to claim 1 characterized by deleting the timer queue element held next from this timer queue means with this minimum timer queue element

[Claim 12] The timer management method which is characterized by providing the following and which manages a hawksbill turtle vent by the queue. This timer management method is a step which holds the timer queue element which corresponds to a hawksbill turtle vent and has a timer value according to the deadline time of this hawksbill turtle vent for one or more timer queue meanses by the queue according to the publishing agency of this hawksbill turtle vent. The selection step which chooses the minimum timer queue element which has the minimum timer value in the timer value which the separate timer queue element held at the head of the separate timer queue means about the one or more aforementioned timer queue meanses has when the timer interruption according to the hawksbill turtle vent occurs. The rearrangement step which holds at the head the timer queue element held at the degree of this minimum timer queue element, and deletes this minimum timer queue element from this timer queue means about a timer queue means to hold the aforementioned minimum timer queue element. The reconfiguration step which newly reconfigures the value which subtracted the minimum aforementioned timer value from the timer value which each timer queue element held at the head of each timer queue means has about other different timer queue meanses from a timer queue means by which the aforementioned minimum timer queue element was held as a timer value.

---

[Translation done.]

\* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

## DETAILED DESCRIPTION

---

### [Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] this invention relates to the timer management equipment and the method of performing timer control of the system from which plurality differs with one timer, and using timer resources efficiently in timer management equipment and a method especially communication, measurement, a control system, etc.

[0002]

[Description of the Prior Art] There were mainly three methods in the timer management method currently used when performing system controls, such as scheduling, by communication, measurement, control, etc. conventionally, or when a real time operating system (OS) performed task management. These three methods are explained in order below.

[0003] (1) The method using an event timer managed table.

One system consists of two or more batches (task) divided by the function, and one task has two or more operating state.

[0004] Drawing 17 shows the state transition of a task. As shown in drawing 17, one task has two or more operating state, and there are three states, a running state (RUN) 1700, the execution waiting state (READY) 1710, and dormant state (WAIT) 1720, in this operating state. According to the situation of operation, each task is between a running state 1700 and dormant state (1725 1735) between dormant state 1720 and the execution waiting state 1710 (1745 1755) and between the execution waiting state 1710 and a running state (1715 1705), and changes the operating state (changes). In a subsystem, the number of tasks of a running state (RUN) 1700 is surely 1. If events, such as interruption, occur, among two or more tasks in a running state 1700 or the execution waiting state 1710, a task with the highest priority will be chosen by OS (dispatch), and will carry out changes etc. to a running state (1715). The task of a running state 1700 changes to dormant state 1720 to carry out waiting for an event itself, when the right of execution is taken by other tasks, without being dispatched. The task of dormant state 1720 changes to an execution waiting state or a running state, when the rising directions from the task in generating or the running state of an event are received (1745 1735).

[0005] The conventional method of managing the identification information about an above-mentioned task etc. is explained below. Drawing 18 shows the timer managed table of the hawksbill turtle vent in the conventional realtime OS at the time of considering a task to be a timer user. In drawing 18 (A), the dispatching priority of Task ID (task number) and a task, the stack pointer for every task, etc. are recorded on the identification information 1800 of the timer user in the timer managed table 1850 (task). The timer value 1810 expresses \*\*\*\*\* of a working timer. The task with the larger timer value 1810 than 0 is the waiting for a hawksbill turtle vent, and the operating state 1820 of a task is dormant state (WAIT). A task with the larger timer value 1810 than 0 does not acquire the right of execution of CPU. The timer value 1810 of the operating state 1820 of the task of 0 is in a running state (RUN) 1700 or the waiting state (READY) 1710 waiting for execution. Although a task emits a system call and goes into dormant state 1720, one task waits for two or more hawksbill turtle vents simultaneously, and it does not go into dormant state 1720. Namely, as a task escapes from dormant state 1720, the number of the hawksbill turtle vents to which starting is applied shall be one, therefore one hawksbill turtle vent shall exist for every task. A hawksbill turtle vent (timer interruption) shall be generated for every fixed time. In the case of many OS's, this fixed time is about about 10ms. When the publishing agency task or hawksbill turtle vent which published and registered the hawksbill turtle vent occurs and it is OS which has the function which notifies generating of a hawksbill turtle vent with a certain signal to the notice place task which receives the notice, a task does not need to wait for generating of a hawksbill turtle vent, and does not need to change to dormant state 1720. In the case of such an OS, a task can wait for generating of two or more hawksbill turtle vents simultaneously, without changing to dormant state 1720. Generally, the notice of generating (henceforth "deadline") of a hawksbill turtle vent is performed by the mail box. As an example which waits for generating of two or more hawksbill turtle vents Although the hawksbill turtle vent A will be deleted as normal termination if the processing which is before the 1st hawksbill turtle vent A occurs can be ended when turning ON the power supply of equipment by the 1st generating of a hawksbill turtle vent and turning OFF the power supply of equipment by the 2nd generating of a hawksbill turtle vent or When halt processing is started and the 2nd hawksbill turtle vent B occurs noting that it is unusual, if the hawksbill turtle vent A occurs, there is carrying out an emergency shut down as what failed also in halt processing etc.

[0006] Drawing 18 (B) shows a timer managed table in case a task waits for generating of two or more hawksbill turtle vents as mentioned above. As shown in drawing 18 (B), the timer managed table 1800 serves as structure (henceforth a "timer queue") which connected the cell 1830 or the cell 1835 in the shape of a queue, and the start address (pointer) to a TAINA queue is put on the portion of the timer value 1810 (sign 1825 of drawing 18 (B)) of drawing 18 (A). Thus, the actual timer is connected to the timer queue.

[0007] Drawing 19 shows the example of the system which consists of two subsystems, the system 1 (1900) for explaining the conventional timer interruption processing, and a system 2 (1910), and drawing 20 shows the flow chart of the conventional timer interruption processing. Below, processing of the conventional timer interruption is explained using drawing 19 and drawing 20. As shown in drawing 19, a timer user's (task) identification information (task ID) 1800 and the timer value 1810 grade are recorded on the timer managed table 1920 for every each system 1900 and ] 1910. Each systems 1900 and 1910 perform hawksbill turtle vent registration 1905 to the timer managed table 1920, and deadline 1915 is notified to each systems 1900 and 1910. The timer value 1810 of the timer table 1920 is processed by the interruption 1925 (usually periodic interruption for about 10ms) from timer equipment 1930. Generating of a timer interrupt 1925 reduces one timer value 1810 of all tasks with the larger timer value 1810 than 0 (a decrement is carried out). The timer value 1810 carries out nothing from the start to the task of zero. This processing is performed about all the tasks registered (Step S202 of drawing 20, or S208). The state of a task is made to change from the hawksbill turtle vent waiting state (WAIT) 1720 to the execution waiting state (READY) 1710 in processing of a timer interruption 1925 to the task which only 1 carried out the decrement of the timer value 1810, and was set to 0, i.e., the task from which timer

\*\* 1810 changed to 1->0, (Steps S208 and S209). In a timer interrupt 1925, if all decrement processings (Step S206) of the timer value 1810 are judged to be ends (Step S202), it will return to dispatch processing of realtime OS etc. (Step S203). When processing of a timer interrupt 1925 was completed as a part of processing of OS in dispatch processing, it is \*\*\*\*\* about the right of execution of CPU to a task with the highest dispatching priority. Specifically, the stack-pointer information (context of a task (context:executable-statement pulse)) in a timer user's (task) identification information 1800 is restored, namely, it is set as the stack pointer register of CPU, and execution is moved to a task with the highest dispatching priority.

[0008] Above-mentioned task interruption processing is explained more to a detail using drawing 20 . Generating of a timer event (timer interrupt) performs the following operation from the head of the timer tables 1850 or 1920. A timer judges whether it is under [ use ] \*\*\*\*\* after end judgment (Step S202) of initial setting (Step S200) and all hawksbill turtle vents (Step S204). The case where a timer is using it is a case where the timer value 1810 in timer table 1850 grade is one or more. When a timer is using it, the timer value 1810 is carried out minus 1 (Step S206). Deadline processing is performed when the timer value 1810 is set to 0 (Steps S208 and S209). The deadline processing S209 is processing which notifies generating of a task HETAI my vent which registered the hawksbill turtle vent as mentioned above. As a notice method of a hawksbill turtle vent, a mail box is used as mentioned above, or an event flag is stood into timer table 1850 grade, and there is the method of making OS process, and being [ the method etc. / it ] sufficient and setting to it etc. After processing of all hawksbill turtle vents is completed, it jumps to the dispatcher of OS.

[0009] As mentioned above, it is possible to manage two or more tasks by one timer also in the conventional method. However, by the conventional method, to the system which consists of two or more subsystems, two or more subsystems will be intermingled on the timer table 1920, and there was a problem that the management method between two or more systems will become very complicated. That is, a different system in one timer managed table may be intermingled, and the operational status of a certain system might affect management of the timer of another alien system. A system while \*\*\*\* for which one system which furthermore exists interrupted execution, and the interrupted system are also working will also be intermingled on the same timer table 1920. Since the deadline of the timer of the system which has interrupted operation cannot be made to pass, decrement processing S206 in the timer table 1920 must be performed only to the task of the system under operation, and the timer table 1920 corresponding to the system under a halt or discontinuation must perform processing of not making it change. Thus, in the conventional method, since timer interruption was separately prepared for two or more systems of every, there was a problem that the management method between two or more systems will become very complicated.

[0010] (2) How to use a timer queue (difference timer queue).

the difference for which drawing 21 is used conventionally -- the data structure of a timer queue is shown As shown in drawing 21 , the queue structure (timer queue) 2100 is formed for the timer value 2016 grade of generating time intervals, such as the hawksbill turtle vents 1 (2118), such as the hawksbill turtle vent 1 (2118) of each system, and the pointer (chain information) 2114 grade to the hawksbill turtle vent of the following system as one cell 2110 grade. For example, a cell 2110 is a cell of a system 1 and, next, the cell 2120 of a system 2 is connected. It connects with the head queue 2102, the head cell 2110 is vacant at the empty queue 2104, and the head cell of a cell (un-illustrating) is connected.

[0011] drawing 22 -- the setting method of the timer queue 2100 -- letting it pass -- difference -- operation of a timer is shown In drawing 22 , a horizontal axis is a time-axis and sets all event generating time standard time to zero on a time-axis. A vertical axis shows four waiting hawksbill turtle vents for time (70ms, 100ms, 150ms, 200ms). These four hawksbill turtle vents made operation start simultaneously. The time measurement unit (time resolution) of hardware is set to 10ms. The deadline of four hawksbill turtle vents 1 of drawing 22 been and shown or 4 is passed sequentially from the small thing of the setup time. For example, 200ms hawksbill turtle vent of the hawksbill turtle vent 4 does not pass the deadline of previously from 70ms hawksbill turtle vent of the hawksbill turtle vent 1. Four hawksbill turtle vents 1 or 4 is generated on a time-axis at intervals of 70ms, 30ms, 50ms, and 50ms. That is, each generating time is the time of TE1 (70ms), TE2 (100ms), TE3 (150ms), and TE4 (200ms). Therefore, the timer value 2116 of an event 1 (2118) connects the cell 2110 for 70ms to the head of the timer queue 2100. TE2-TE1=100-70=30ms which is the difference of the generating time of the hawksbill turtle vent 2 (2128) and the hawksbill turtle vent 1 (2118) is set to the timer value 2126 of the cell 2120 linked to the 2nd. this difference -- the thing of a value -- difference -- it is called a timer value The same is said of 3rd henceforth. since 70ms passes, and further 30ms will pass if the 2nd cell 2120 passes the deadline of from there, when the cell 2110 of an eye passes the deadline of most by carrying out like this, until the cell 2120 of No. 2 passes the deadline of from time zero (at the time of timer issue) -- 70+30=100 ms -- progress -- the bottom -- alias -- it becomes Similarly, the cell 2140 of the 2130 or 4th cell of No. 3 can realize necessary timer operation, if the difference from the deadline time of the cell in front of one is set up as timer value 2126 grade.

[0012] The calculation method of the numeric value in drawing 22 is shown in the following table 1.

[0013]

[Table 1]

タイマイイベント	タイムアップ時間 (m s)	タイマ設定値	差分値
タイマイイベント 1	70 m s	7	70 - 0
タイマイイベント 2	100 m s	3	100 - 70
タイマイイベント 3	150 m s	5	150 - 100
タイマイイベント 4	200 m s	5	200 - 150

[0014] In Table 1, deadline time shows the deadline time (a start time is set to 0) for every event. the timer value with which the set point is set as a timer queue -- being shown -- difference -- a value shows relative time

[0015] the difference of the former [ drawing 23 ] -- the flow chart of processing using the timer queue is shown In drawing 23 , it is confirmed whether one or more hawksbill turtle vents effective in the timer queue 2100 are connected (Step S230). A thing with the larger (timer value > 0) timer value of each cell (henceforth a "timer queue element") which is the element of the timer queue which "is effective" here than 0 is said. An effective timer queue element ends processing, when one is not connected. When that is not right, a signal is sent to the task which is waiting for the hawksbill turtle vent and which it has (Step S232). If the timer queue element of the head whose deadline was passed is removed from a timer queue and it connects in other timer queue elements after that, the timer queue element will be registered into the head of

a timer queue managed table (Step S236). The timer value of the timer queue element of the timer shaft head is set as a hardware timer (Step S238).

[0016] As mentioned above, it is possible to manage two or more tasks by one timer also in the conventional method. However, by the conventional method, to the system which consists of two or more subsystems, a different system in one timer queue may be intermingled, and the operational status of a certain system might affect management of the timer of another alien system. Since a system while \*\*\*\* for which one certain system interrupted execution, and the interrupted system are also working will also be intermingled in the same timer queue, if a certain system is stopping, a head queue will not necessarily pass the deadline of. Therefore, it needed to look for the system under operation. Since the deadline of the timer of the system which has interrupted operation was not able to be made to pass, removal processing from the queue in a timer queue was performed only to the task of the system under operation, and the hawksbill turtle vent corresponding to the system under a halt or discontinuation had the problem that processing of not making it change had to be performed. Therefore, in the conventional method, since preparation etc. made the timer (interruption) separate for two or more systems of every, there was a problem that the management method between two or more systems will become very complicated. Furthermore, since invalid data (data corresponding to the system under pause) will have connected into a timer queue when it has stopped the whole subsystem (for example, when it changes from 2 system operation to 1 system operation), the processing (dequeue operation) removed from a timer queue becomes complicated. Only for the chain information just behind a certain timer queue element, operation which carries out a dequeue from a timer queue was not completed, but there was a problem that the timer queue element of the pause system which follows chain information by a certain method as a result, and corresponds had to be looked for.

[0017] (3) The task management table to a periodic-start event.

The conventional management method for a hawksbill turtle vent to perform operation and starting for every fixed time like fixed surveillance operation for every fixed time, for example, state check operation, is explained.

[0018] Drawing 24 shows the managed table 2450 of the conventional periodic-start task. In drawing 24, periodic-start time etc. is recorded on a timer user's (task) identification information 2400 at the time of initial setting of a periodic-start task. The timer value 2410 is a \*\*\*\*\* timer value. The timer value 2410 shows the task which is not working as for the task of 0. The timer value 2410 is carried out minus 1, whenever regular timer interruption (10ms interruption) enters. It is detected that the timer value 2410 was set to 0, and generating of a hawksbill turtle vent is notified to a task at a moment. The timer value 2410 reconfigures the periodic-start time of a timer user's (task) identification information 2400. If the task to which generating of a hawksbill turtle vent was notified performs task operation and it ends, it will return to the execution waiting state 1710. That is, the right of execution is abandoned and it jumps to a dispatcher.

[0019] Drawing 25 (A) and (B) show operation of the task in the conventional task management method. Although two or more periodic-start tasks may exist, they are asynchronous to each other. As shown in drawing 25 (A), TE10 and TE20 are the starting time of a periodic-start task, the periodic-start time 2500 is TE20-TE10=15-5=10ms in this example, and the task operating time 2550 is TE15-TE10=10-5=5ms. Thus, since task operation is completed by the task time 2550 shorter than the periodic-start time 2500 even if the timer value 2410 is set up working [ a periodic-start task ] in the case of the periodic-start time 2500> task operating time 2550 and the decrement is carried out every [ 1 ], no problem is generated in operation of a system. Since the 2nd task starting time is the time of TE25, and it is the periodic-start time 2500< task operating time 2600 when the task operating time 2600 is TE25-TE10=20-5=15ms as shown in drawing 25 (B) on the other hand, the task started at periodic-start time is not completed within the periodic-start time 2500. Therefore, in the time of TE20, a task will be started doubly, and there was a problem that a bad influence appeared in operation of a system.

[0020] By the conventional method, the managed table 2450 for the periodic-start task of displaying operating state for every periodic operation, for example, ls, needed to be prepared as mentioned above apart from the timer table 1850 grade for managing a hawksbill turtle vent. Periodic-start operation was processed by the managed table 2450 grade which became independent of timer table 1850 grade and which became independent as a function. That is, there was a problem that two kinds of timer table management methods were required. Thus, in the conventional timer management method, the periodic-start hawksbill turtle vent was separated and managed independently with the one-shot type hawksbill turtle vent processing which is not periodic. Furthermore, since periodic-start hawksbill turtle vents were operating independently, even if it was a periodic-start event with the same periodic start, when both had the time phase contrast which is a difference of the timing which a hawksbill turtle vent generates, the concurrence (simultaneous processing) of a hawksbill turtle vent was not able to be made to perform. For this reason, there was a problem that it was in the inclination for the number of times of timer interruption to increase, and there was an inclination for the consumed electric current of the system by timer interruption to increase. For example, if nothing is not carried out even if it is under telephone call since a line of contact is held in hardware even if it is under telephone call when a system is a cellular-phone system, CPU will save the power by changing to dormant state (WAIT). That is, power-saving is attained by carrying out the sleep (WAIT) not only of a task but the CPU itself. Therefore, since power consumption would increase if CPU is started by timer interruption etc., when the number of times of this timer interruption increased, there was a problem that power consumption increased certainly. Furthermore, in the conventional timer management method, when time longer than the time measurement range set up in hardware was planned, apart from the timer management on hardware, the counter with large bit size was prepared and it corresponded to the hawksbill turtle vent with software for a long time. For this reason, when the time basis and time measurement range on hardware changed, count processing of the timer by software was also influenced. Furthermore, in the conventional timer management method, there is no concept of the allowable error at the time of time measurement, and the hawksbill turtle vent was generated with the end (deadline) of the timer time simply set up in hardware. For this reason, dequeue processing was not able to be carried out though the hawksbill turtle vent which becomes in a time measurement top tolerance was connected even if it made the near pass the deadline of simultaneously (just after) when a certain hawksbill turtle vent passed the deadline of. That is, in the conventional timer management method, there was a problem that there was no function which unifies a nearby event very much in time.

[0021]

[Problem(s) to be Solved by the Invention] As mentioned above, since two or more subsystems were made intermingled on one timer table to the system which consists of two or more subsystems, there was a problem that the management method between two or more systems will become very complicated in the conventional timer management method. A different system in one timer queue of what can manage two or more tasks by one timer may be intermingled, and the operational status of a certain system might affect management of the timer of another alien system. Furthermore, the managed table for a periodic-start task needed to be prepared apart from the timer table for managing a hawksbill turtle vent. There was a problem that it is in the inclination it to come gaily to make the concurrence of a hawksbill turtle vent perform, and for the number of times of timer interruption to increase since there was nothing when there is time phase contrast, even if it is a periodic-start event with the same periodic start, and there was an inclination for the consumed electric current of the system by timer interruption to increase. Although it furthermore corresponded to the hawksbill turtle vent for a long time using software by limit of the

number of bits of a hardware timer, when hardware up conditions changed, processing by software was also influenced. Since there was no concept of the allowable error at the time of time measurement, there was a problem that there was no function which unifies a nearby hawksbill turtle vent very much in time.

[0022] Then, the purpose of this invention is by being made in order to solve the above-mentioned problem, and managing the hawksbill turtle vent of two or more systems unitary to offer the timer management equipment and the method system operation from which plurality differs with one timer equipment is realizable. Other purposes of this invention are by treating a periodic-start hawksbill turtle vent like an one-shot type hawksbill turtle vent to offer the timer management equipment and the method of management of a periodic-start hawksbill turtle vent and an one-shot type hawksbill turtle vent being unifiable, being able to make the concurrence of a hawksbill turtle vent able to perform, even if it is the case where there is time phase contrast further, being able to reduce the number of times of timer interruption, and cutting down the consumed electric current of the system by timer interruption. Other purposes of this invention are by forming a hawksbill turtle vent into a periodic-start hawksbill turtle vent for a long time to offer the timer management equipment and the method of treating a hawksbill turtle vent within the number of bits of a hardware timer for a long time. Furthermore, other purposes of this invention are by using the precision of a timer value to offer the timer management equipment and the method of unifying a nearby hawksbill turtle vent very much in time.

[0023]

[Means for Solving the Problem] The timer management equipment of this invention is timer management equipment which manages a hawksbill turtle vent by the queue. this timer management equipment They are one or more timer queue meanses to hold the timer queue element corresponding to a hawksbill turtle vent by the queue according to the publishing agency of a hawksbill turtle vent. That in which this timer queue element has the difference of the deadline time of a hawksbill turtle vent, and the time connected to this timer queue as a timer value, When timer interruption occurs, about the one or more aforementioned timer queue meanses A selection means to choose the minimum timer queue element which has the minimum timer value in the timer value which the separate timer queue element held at the head of the separate timer queue means has, A rearrangement means to hold at the head the timer queue element held at the degree of this minimum timer queue element, and to delete this minimum timer queue element from this timer queue means about a timer queue means to hold the aforementioned minimum timer queue element, About other different timer queue meanses from a timer queue means by which the aforementioned minimum timer queue element was held It has a reconfiguration means to newly reconfigure the value which subtracted the minimum aforementioned timer value from the timer value which each timer queue element held at the head of each timer queue means has as a timer value. The separate timer queue managed table means respectively corresponding to [ here ] the timer queue means of the aforementioned exception individual in the timer management equipment of this invention, It has further a system queue managed table means to manage the timer queue managed table means of the aforementioned exception individual in predetermined order. the aforementioned selection means This aforementioned timer queue managed table means separate in predetermined order is chosen, and the minimum timer queue element is chosen about the separate aforementioned timer queue means corresponding to the separate aforementioned selected timer queue managed table means. Here, when the operating state of the issue origin of the hawksbill turtle vent corresponding to [ equipment / timer management / of this invention ] the aforementioned timer queue managed table in the aforementioned system queue managed table means changes, other different timer queue managed tables from the aforementioned timer queue managed table are managed independently. Here, when an issue origin's of hawksbill turtle vent corresponding to [ equipment / timer management / of this invention ] aforementioned timer queue managed table in aforementioned system queue managed table means existence state changes, other different timer queue managed tables from the aforementioned timer queue managed table are managed independently. Here, the aforementioned minimum timer queue element with which the timer management equipment of this invention was deleted when the aforementioned rearrangement means corresponded to the periodic hawksbill turtle vent by which the aforementioned minimum timer queue element deleted from the aforementioned timer queue means is started for every predetermined periodic time is re-connected to the position corresponding to this predetermined periodic time [ in the aforementioned timer queue means ] after. The timer management equipment of this invention is further equipped with the connecting means which newly connect the timer queue element corresponding to a hawksbill turtle vent to the aforementioned timer queue means here. these connecting means In the state where the 1st timer queue element corresponding to the periodic hawksbill turtle vent started for every 1st period time is already connected to the aforementioned timer queue means When the 2nd timer queue element corresponding to the periodic hawksbill turtle vent started for every 2nd period time which has the 1st period time and a predetermined relation is newly connected to the aforementioned timer queue means, Connection which makes this time earliest deadline time of the 1st timer queue element and 1st deadline time of the 2nd timer queue element is made. 1 time of timer interruption -- the [ the 1st timer queue element and ] -- the number of times which makes the deadline of 2 timer queue element pass simultaneously is increased, and the consumed electric current by timer interruption is cut down The aforementioned predetermined relation with the 1st period time when the 2nd period time has timer management equipment of this invention is a relation whose 2nd period time is the multiple of the 1st period time here. the aforementioned connecting means The value which lengthened the present time from the earliest time whose deadline the 1st timer queue element passes is connected as 1st timer value of the 2nd timer queue element, and deadline time of the 1st timer queue element and deadline time of the 2nd timer queue element are made into this time. Here, the aforementioned predetermined relation with the 1st period time when the 2nd period time has timer management equipment of this invention is a relation whose 2nd period time is the divisor of the 1st period time, and the aforementioned connecting means connect the 2nd timer queue element at the earliest time when the time which lengthened the integral multiple of the 2nd period time from the earliest time whose deadline the 1st timer queue element passes does not become later than the present time. Here, the aforementioned predetermined relation with the 1st period time when the 2nd period time has timer management equipment of this invention is a relation with the as relatively prime 2nd period time as the 1st period time, and the aforementioned connecting means are connected so that the 1st timer queue element and the 2nd timer queue element may pass the deadline of simultaneously for every period time used as the least common multiple of the 1st period time and the 2nd period time. Here, the timer management equipment of this invention is further equipped with the connecting means which newly connect to the aforementioned timer queue means the timer queue element corresponding to the hawksbill turtle vent which has predetermined warm-up time, and these connecting means have the number of times repeatedly with the periodic time based on the aforementioned predetermined warm-up time for the aforementioned timer queue element, and it changes and connects them to the periodic timer queue element corresponding to this periodic hawksbill turtle vent started by the number of times of a repeat for every periodic time of this. The timer management equipment of this invention is further equipped with an allowable error [ as opposed to the aforementioned timer value in the aforementioned timer queue element ] here. the aforementioned rearrangement means When the aforementioned allowable error which the timer queue element held at the degree of this minimum timer queue element has about a timer queue means to hold the aforementioned minimum timer queue element is smaller than the value which subtracted the minimum aforementioned timer value from the aforementioned timer value, this -- the timer queue element held next is deleted from this timer queue means with this minimum timer queue element

[0024] The timer management method of this invention is a timer management method which manages a hawksbill turtle vent by the queue. This timer management method The step which holds the timer queue element which corresponds to a hawksbill turtle vent and has a timer value according to the deadline time of this hawksbill turtle vent for one or more timer queue means by the queue according to the publishing agency of this hawksbill turtle vent, When the timer interruption according to the hawksbill turtle vent occurs, about the one or more aforementioned timer queue means The selection step which chooses the minimum timer queue element which has the minimum timer value in the timer value which the separate timer queue element held at the head of the separate timer queue means has, About a timer queue means to hold the aforementioned minimum timer queue element, the timer queue element held at the degree of this minimum timer queue element is held at the head. The rearrangement step which deletes this minimum timer queue element from this timer queue means, About other different timer queue means from a timer queue means by which the aforementioned minimum timer queue element was held It has the reconfiguration step which newly reconfigures the value which subtracted the minimum aforementioned timer value from the timer value which each timer queue element held at the head of each timer queue means has as a timer value.

[0025]

[Embodiments of the Invention] Hereafter, with reference to a drawing, the form of operation of this invention is explained in detail. Drawing 1 or drawing 5 shows the data structure used in common in the form 1 of operation of this invention, or 5. The data structure shown in drawing 1 or drawing 5 realizes two or more systems with one hardware timer, two or more time precision, two or more operation, for example, one-shot operation, cycling, prolonged operation, etc. One-shot operation is operation of a hawksbill turtle vent (henceforth a "one-shot type hawksbill turtle vent") which operates only once in a timer queue here. Cycling is operation of a hawksbill turtle vent (henceforth a "periodic hawksbill turtle vent") which carries out multiple-times repeat operation in a timer queue. Prolonged operation is operation of a hawksbill turtle vent (henceforth a "long-term hawksbill turtle vent") with the number of counts of a hardware timer comparatively longer than other hawksbill turtle vents in a timer queue. In order to realize these operation, time phasing between the timer queues with which the generating timing of time phase doubling in the timer queue with which the generating timing of the hawksbill turtle vent within one system is doubled, and the hawksbill turtle vent between two or more systems is doubled is performed using two or more division-into-equal-parts timer queues.

[0026] Drawing 1 shows the structure of the timer queue for every subsystem. As shown in drawing 1, the queue structure (timer queue) 100 is formed for the timer value 116 grade of generating time intervals, such as the hawksbill turtle vents 1 (118), such as the hawksbill turtle vent 1 (118) of a subsystem, and the pointer (chain information) 114 grade to the following hawksbill turtle vent as one cell (timer queue element) 110 grade. For example, the timer queue element 120 is connected to the degree of the timer queue element 110. In drawing 1, although not illustrated, a timer queue is bidirectional linear-list structure. The head timer queue element 110 is connected to the head queue 102, at the empty queue 104, it is vacant, and the head cell of a cell (un-illustrating) is connected.

[0027] Drawing 2 shows the data structure of a timer queue element. As for the front link 202, in drawing 2, the pointer to the back timer queue element in a timer queue and the timer ID 206 of the pointer to the front timer queue element in a timer queue and the back link 204 are [the identifier of a hawksbill turtle vent therefore the timer queue element concerned, and timer classification and precision 208] timer values which set up the working bit width of face of a hardware timer or working timer count units (resolution, precision, etc.), and the timer value 210. The timer value 210 -- this specification -- setting -- difference -- although a timer value is used -- this invention -- difference -- it is not limited to a timer value ID of the object to which the notice place information 212 furthermore published the hawksbill turtle vent in drawing 2, The timer operation classification 214 one-shot operation, cycling, or for a long time The classification of one-shot operation (prolonged operation and one-shot operation), The periodic-start time 216 for a long time which performs one-shot operation the periodic-start time of a periodic hawksbill turtle vent, or for a long time which performs cycling The initial value of a hawksbill turtle vent (as shown by the form 4 of the below-mentioned operation, a hawksbill turtle vent is formed into a periodic hawksbill turtle vent for a long time), The periodic-start counter 218 is the working-system classification for the number of times of a repeat of a hawksbill turtle vent and the system classification 220 detecting incorrect connection a periodic-start hawksbill turtle vent or for a long time. The information about the timer queue element of others of a pan can also be set to the timer queue element 200.

[0028] A timer queue managed table is in front of the head timer queue element of a timer queue. The head queue [in / drawing 1 / in this timer queue managed table] 102 corresponds.

[0029] It connects in front of a timer queue managed table, and drawing 3 shows the data structure of the timer managed table which manages a timer queue. A timer managed table is prepared for every system. The pointer to the system in front of the system with which the pointer 302 to a front system is managed on the timer managed table 300 concerned in drawing 3 (link), The pointer to the next system of the system with which the pointer 304 to the following system is managed on the timer managed table 300 concerned (link), The pointer 306 to a head queue The pointer to the timer queue managed table of the system concerned, The flag for the operating state 308 of a system displaying the operating state of working [of a system] and the system of the halt middle class or the status, and the system classification 310 are working-system classification. The information about the system concerned of others of a pan can also be set to the timer managed table 300.

[0030] Drawing 4 shows the data structure of the system queue managed table which manages the timer managed table 300 for every system. In drawing 4, the link information 402 to a system is a pointer to an alien system, when each system which the system queue managed table 400 manages is made into a subsystem and the system which consists of these subsystems is used as one system as a whole. Furthermore in drawing 4, the state of working [of the system concerned when the pointer to an empty timer queue and the operating state 408 of a system use the pointer to the timer managed table 300 of a head (factice) system and the pointer 406 to an empty queue as one system as a whole in the pointer 404 to a head queue], under a halt, or the discontinuation middle class, the other information on everything [management information] but the system concerned, and the system classification 412 are the working system classification concerned.

[0031] Drawing 5 shows the relation of the control structure of the above-mentioned system queue managed table 400, the timer managed table 300, a timer queue managed table, a timer queue, etc. As shown in drawing 5, the timer queue element 535 grade of one system (factice) is respectively connected in the shape of a queue, and this queue is managed by timer queue managed table 532 grade for every system (factice). Although the whole 540 grade containing this timer managed table 532 grade is pointed out when calling it a timer queue, the queue which only begins from the timer queue element 535 may be pointed out. (Factice) A system is managed by the timer managed table 530 grade for every system (factice), and a \*\* (factice) system is managed on the system queue managed table 510. (Factice) As an example of a system, there is a sign division multiplex (Code Division Multiple Access : CDMA) system, a digital cellular phone / car telephone (Personal Digital Cellular : PDC) system, or PHS (Personal Handyphone System). The timer queue element which the timer counted up the empty timer queue 520 and processing ended, i.e., the timer queue element which it finished using, is connected. If an empty timer queue element is lost, since a new hawksbill turtle vent cannot be published, it will perform garbage collection etc. suitably. the case where timer queue element 535 grade is in timer queue 540 grade when connecting a new timer queue element to timer queue 540 grade -- difference -- a timer value is calculated and a new timer queue element is connected to a timer queue On the other hand, when there is no timer queue element, the timer value 210 is set as

the head of timer queue 540 grade as it is, and a new timer queue element is connected as it is.

[0032] The form 1 of form 1. implementation of operation shows the timer management equipment and the method operation of the system from which plurality differs with one timer equipment (factice) is realizable by unifying the timer queue of two or more systems (factice), and managing the hawksbill turtle vent of two or more systems (factice) unitary. Especially in the following, if it is called a system unless it is shown clearly, each system managed in timer managed table 530 grade will be pointed out. As a whole, when [ which attains to a relation with one system ] managed on the system queue managed table 510, one system is called system as a whole, and each system managed in timer managed table 530 grade is called subsystem. With the form 1 of operation, when the time phase doubling method and hawksbill turtle vent between the timer queues for every system occur, operation of removing the head timer queue element of the system corresponding to this hawksbill turtle vent is explained as a center. The data structure shown by drawing 1 or 5 is used for control data structure.

[0033] Drawing 6 shows the timing diagram in the state where timer interruption occurred at Time Ti, when three systems (factice) 1 or 3 exists. Each system 1 or 3 has the timer queue 540 in drawing 5, or 580 respectively, and each timer queue 540 grade is independent for every [ each system 1 or ] three. In drawing 6, when timer interruption occurs at Time Ti, the timer queue element which compares the timer queue element 535 grade of the head of the timer queue 540 grade for every system, and has the minimum timer value 210 is chosen. In the example of drawing 5, since the timer value 210 of the head timer queue element 535 of a system 1 is [ the timer value 210 of 7 and the head timer queue element 575 of a system 3 of the timer value 210 of 10 and the head timer queue element 555 of a system 2 ] 3, 3 of the timer value 210 of a system 3 is the minimum. Therefore, it can be judged that the head timer queue element 575 of a system 3 passed the deadline of. Then, the head timer queue element 575 of a system 3 is removed from the timer queue 580, and let the following timer queue element 577 (the timer value 210 is 2) be the head timer queue element of a system 3. The removed timer queue element is connected to the empty timer queue 520. The operation at the time of deadline of the timer queue element [ in / a system 3 / as mentioned above ] 575 is completed. Although the deadline of the timer queue elements 535 and 555 of a system 1 and a system 2 is not passed at this time, it is necessary to lessen deadline time of each head timer queue element by time progress of 3 which is the timer value 210 of the head timer queue element 575 of a system 3. Then, the value which subtracted 3 of the timer value 210 of the timer queue element 575 of the system 3 whose deadline was passed from each head timer queue elements 535 and 555 of systems 1 and 2 is reconfigured as a timer value 210. In the case of a system 1, in the case of 10-3=7 and a system 2, the timer value 210 turns into a timer value with 7-3=4 [ new ], and the timer value 210 serves as the content of the timer value 210 of the head timer queue elements 535 and 555 respectively. After reconfiguring each system 1 or the timer value 210 of the head timer queue element of 3, the minimum thing of the value is set as a hardware timer, and timer table processing is completed. Since the timer values 210 of a system 3 are 2 and the minimum in this example, 2 is set to a hardware timer. In this example, although only the deadline of one timer queue element 575 was passed, depending on the timer value 210 set up, the timer queue element of two or more systems may pass the deadline of simultaneously. In this case, all the timer queue elements whose deadline was passed perform processing removed from timer queue 540 grade.

[0034] Drawing 7 shows the processing in the gestalt 1 of this operation with a flow chart. In drawing 7, the value of the head timer queue element 535 grade of timer queue 540 grade is compared between each system. However, by the operating state 308 of the system of timer managed table 530 grade, a system does not process to the timer queue 540 grade which is not operating by the case where there is no timer queue element effective in timer queue 540 grade etc., during a halt when not mounted. In this case, the pointer 304 to the following system of timer managed table 530 grade is followed, and the timer queue 560 grade of the following system is processed. The system classification 310 is recorded simultaneously (Step S70). Since the timer value 210 of head timer queue element 575 grade is the timer queue element whose deadline the minimum timer queue element 575 grade passed when a timer passes the deadline of, it removes from this timer queue element 575 grade timer queue 580 grade. When two systems pass the deadline of simultaneously (i.e., when the value of head timer queue element 575 grade is the same), two timer queue element 575 grades are removed. When three or more are the same, it removes similarly. Comparison of the timer value 210 between above-mentioned systems is performed by following the pointer 304 to the following system of timer managed table 530 grade. Therefore, the art of the gestalt of this operation is not influenced with the number of the subsystems which constitute a system. Since the system is only skipped in case the pointer 304 to the following system is followed and it is searched even if the subsystem of a system which exists working is stopped or deleted, processing of other working subsystems is not affected. Next, the timer value 210 of the head timer queue element 535 grade of all subsystems is amended. The zero clear of the system number counter i is carried out as initial setting (Step S72). Processing of S79 is repeated from Step S76 until the processing to all subsystems is completed (Step S74). First, it judges whether it is the subsystem whose deadline the subsystem passed (Step S76). Since adjustment of the timer value 210 is unnecessary if it is the subsystem whose deadline was passed, it flies to Step S79. the difference of the timer queue element 575 grade which passed the deadline of and was removed from the timer queue 580 grade when that was not right -- the difference of the head timer queue element 555 grade of the subsystem to which others did not pass the deadline of the timer value 210 -- it lengthens from the timer value 210 (Step S78) The time phase of timer queue 540 grade - between subsystems can be doubled by this decrement. In order to adjust the timer value 210 of the head timer queue element 535 grade of the following subsystem, 1 is added to the system number counter i (Step S79). When it returns to Step S74 and completes about all subsystems, the value of the inner minimum of the timer value 210 of head timer queue element 555 grade is looked for, and it is set as a hardware timer.

[0035] In the form 1 of above-mentioned operation, when deleting the whole subsystem, it can realize only by connecting head timer queue element 535 grade with the system queue managed table 510, and emptying the contents of the timer queue managed table 532 grade. Conversely, when extending a system newly, an independent new timer queue, a timer queue managed table, and a timer managed table are generated, and it adds to the list of timer managed tables of other subsystems on which the system queue managed table 510 manages this timer managed table, and can realize only by carrying out, while operating the operating state 308 of the system in which the operating state of a new system is shown. Therefore, a new system can be extended, without affecting other existing portions at all.

[0036] As mentioned above, according to the form 1 of operation, operation of the subsystem from which plurality differs by one hardware timer is realizable by managing the hawksbill turtle vent of two or more subsystems unitary using the timer queue for every subsystem.

[0037] The form 2 of form 2. implementation of operation shows the timer management equipment and the method of unifying management of \*\*, a periodic hawksbill turtle vent, and an one-shot type hawksbill turtle vent treating a periodic hawksbill turtle vent like a general one-shot type hawksbill turtle vent. Hereafter, the form 2 of this operation is explained using drawing 8 or 11. The data structure shown by drawing 1 or 5 is used for control data structure. In the timer queue 540 grade, in order to treat simultaneously the one-shot type hawksbill turtle vent which operates only once, and the periodic hawksbill turtle vent which operates repeatedly with the set point, identification information of operation is prepared as the timer operation classification 214 and periodic-start time 216 into the timer queue element 200 of drawing 2 . The timer queue element whose timer operation classification 214 is cycling (henceforth a "periodic timer queue element") In order to distinguish from this, especially the timer queue element corresponding to an one-shot type hawksbill turtle vent is called one-shot type timer queue element. A

periodic start is carried out every periodic-start time 216.

[0038] Drawing 8 shows operation of the periodic hawksbill turtle vent in the form 2 of operation. Suppose that the periodic timer queue element 802 (periodic-start time 216= 10) of the head of a timer queue passed the deadline of in drawing 8 . Although the periodic timer queue element 802 is removed from the head of a timer queue at this time, since it is again started after the periodic-start time 216 (=10), it re-connects with the position 810 after the periodic-start time 216 (=10) of a timer queue. As shown in drawing 8 , it re-connects after the one-shot type timer queue elements 804 and 806, using the timer value of a periodic start as 3.

[0039] Drawing 9 shows operation of drawing 8 by the timing diagram. As shown in drawing 9 , the time whose deadline the periodic timer queue element 802 passes TA, If time when TC and the one-shot timer queue element 808 pass the deadline of the time when TB and the one-shot timer queue element 806 pass the deadline of the time whose deadline the one-shot timer queue element 804 passes is set to TE It turns out that what is necessary is for the periodic timer queue element 810 which should be re-connected to set the timer value 210 as 3 in the position of the time TD when only the time of 10-2-5=3 has passed since the periodic-start time 216 was 10, and just to re-connect. The timer queue after reconfiguring the periodic timer queue element 810 is shown in drawing 10 .

[0040] Drawing 11 shows processing of the periodic hawksbill turtle vent in the gestalt 2 of operation with a flow chart. In drawing 11 , when a timer passes the deadline of, the notice of an event is performed to timer issue origin (Step S110). This processing is the same processing as the hawksbill turtle vent of one-shot operation. Next, it judges whether it is deadline of a periodic hawksbill turtle vent (Step S112), and if that is right, the periodic hawksbill turtle vent whose deadline was passed will be re-connected to a timer queue (Step S114). When there is no next timer queue element of a periodic timer queue element, a periodic hawksbill turtle vent will be again connected to the head of a timer queue. Finally the timer value 210 of a head timer queue element is set as a hardware timer (Step S116). When it is not deadline of a periodic hawksbill turtle vent at Step S112, it flies to the direct step S116.

[0041] According to the gestalt 2 of this operation, saving of time to look [ to be able to save memory, since the same periodic hawksbill turtle vent does not exist on / two or more / a timer queue, and ] for a timer queue can be aimed at. What is necessary is to delete only one of the periodic hawksbill turtle vent of the from on a timer queue, when stopping a periodic hawksbill turtle vent. Furthermore, according to the gestalt 2 of this operation, even when there is little maximum number of counts (bit width of face of a timer) of a timer in hardware, the restrictions on hardware can be lost by using a timer queue element periodically. For example, about 20 bits of resolution of a timer are needed for the hardware timer with which it realizes the long-term hawksbill turtle vent of this hour since periodic-start time is 100(count/second) = 1 hour =60(minute) x60(part for second) x360,000 count when the longest of the 1 hour is carried out in 10ms. If the periodic timer queue element which this periodic-start time period-izes the long-term hawksbill turtle vent of 1 hour according to the gestalt 2 of operation, for example, makes 1 minute the longest with resolution for 10ms on the other hand is used, it will become 100(count/second) = 1 minute = 60-second x6,000 count. Therefore, the hardware timer which realizes the long-term hawksbill turtle vent of this hour becomes realizable, is about 4 bits less than, and can be managed with bit width of face of 16 bits.

[0042] As mentioned above, since according to the gestalt 2 of operation it can treat like a general one-shot type hawksbill turtle vent when an one-shot type forms a periodic hawksbill turtle vent into a hawksbill turtle vent, management of a periodic hawksbill turtle vent and an one-shot type hawksbill turtle vent is unifiable.

[0043] The gestalt 3 of gestalt 3. implementation of operation shows the timer management equipment and the method of reducing the number of times of deadline of a hardware timer by performing time phase doubling between periodic hawksbill turtle vents. Hereafter, the gestalt 3 of this operation is explained using drawing 12 (A) and (B). The data structure shown by drawing 1 or 5 is used for control data structure.

[0044] Drawing 12 (A) and (B) show operation of the periodic hawksbill turtle vent in the gestalt 3 of operation by the timing diagram. As shown in drawing 12 (A), the periodic hawksbill turtle vent 1 and the periodic-start time 216 in every second shall have [ the periodic-start time 216 ] the periodic hawksbill turtle vent 2 in every 2 seconds. In this case, if time phase doubling is not made, T2, T3, T5, T6, and T8 occurred [ the hawksbill turtle vent 1 ] that the hawksbill turtle vent 1 and the hawksbill turtle vent 2 were asynchronous a total of 5 times, T1, T4, and T7 would occur a total of 3 times, and the hawksbill turtle vent 2 will have occurred a total of 8 times.

[0045] The deadline of two hawksbill turtle vents 1 and hawksbill turtle vents 2 can be made to pass simultaneously to timing T12, T14, and T16 by doubling the time phase which two periodic hawksbill turtle vents 1 and the hawksbill turtle vent 2 generate on the other hand as shown in drawing 12 (B). By this, the number of times of generating of a hawksbill turtle vent can be reduced from 8 times to 5 times. Hereafter, how to double the time phase of two hawksbill turtle vents is explained. As shown in drawing 12 (B), when starting the periodic hawksbill turtle vent 2, (Tz) adjusts the issue time of the hawksbill turtle vent 2 according to the time phase of other periodic hawksbill turtle vents 1 which are on a timer queue and have already moved, and connects with a timer queue. the time of registering the hawksbill turtle vent 2 -- difference with the deadline time T12 and Tz of (Tz) and other periodic hawksbill turtle vents 1 -- deltat=T12-Tz is calculated and this deltat is set up as the first timer value (the 1st time) 210 of the periodic hawksbill turtle vent 2 Namely, timer value 210=deltat, a timer operation classification 214= period, periodic-start time 216 = it sets to time to carry out a periodic start etc. Thus, the timer value 210 of the hawksbill turtle vent 2 is set as deltat instead of the original periodic-start time 216 (=2) only once [ of the beginning ] so that it may synchronize with the generating period (=1) of other periodic hawksbill turtle vents 1. Once it can take a synchronization by T12, as for after, T14 and T16 can take the synchronization. The number of times of the timer interrupt which a hardware timer generates as a result can be reduced. Therefore, since the power which CPU is started by timer interruption etc. and consumed can be reduced, in the case of the cell drive type system of a cellular phone etc., power-saving can be measured greatly.

[0046] For example, the periodic-start time 216 explains operation in case the periodic hawksbill turtle vent of 12 and the hawksbill turtle vent of 72 are added for the periodic-start time 216 to the periodic hawksbill turtle vent of 6. The periodic-start time 216 doubles a time phase with the period of 6, the periodic-start time 216 publishes 12 and warm-up time 216 publishes the hawksbill turtle vent of 72. The timer queue element corresponding to both the hawksbill turtle vent is connected to a timer queue. In this case, since there is the greatest common measure 6 in 6, 12, and 72 of the periodic-start time 216, even if the periodic-start time 216 adds 12 and two hawksbill turtle vents of 72, the number of times started in order that CPU may process a hawksbill turtle vent does not increase. If there is the greatest common measure, there will be no limit in the number of the hawksbill turtle vents added except for conditions, such as memory.

[0047] Conversely, the periodic-start time 216 of it is the same as that of it, when the periodic-start time 216 connects the hawksbill turtle vent of 6 to the event of 72. The periodic-start time 216 is the timing to which it gives the greatest n from which the number with which the periodic-start time 216 subtracted 6xn hours from the time t whose deadline the hawksbill turtle vent of 72 passes since the greatest common measure of 72 and 6 was 6 is set to 0 to it when the periodic-start time 216 connects the hawksbill turtle vent of 6 to the event of 72, and the periodic-start time 216 publishes the event of 6, and should just connect with a timer queue. It considers as the timing which gives the greatest n for considering as the earliest timing.

[0048] An example when the periodic-start time 216 is relatively prime is shown. The periodic-start time 216 shall publish [ the periodic-start

time 216 / the hawksbill turtle vent of 5 ] the hawksbill turtle vent of 3 working. At this time, the event of 3 is relatively prime, namely, there are [ the periodic-start time 216 ] no event and periodic-start time 216 of 5 in a common divisor at a relatively prime number. In this case, the periodic-start time 216 should just adjust the issue timing of the hawksbill turtle vent of 3 so that generating of two hawksbill turtle vents may lap at time for the periodic-start time 216 to serve as the least common multiple 15 (3x5) of 3 and 5.

[0049] As mentioned above, since the number of times of deadline of a hardware timer can be reduced by performing time phase doubling between periodic hawksbill turtle vents according to the gestalt 3 of operation, the power which CPU is started by timer interruption etc. and consumed can be reduced.

[0050] The gestalt 4 of gestalt 4. implementation of operation shows the timer management equipment and the method the prolonged timer more than the maximum number of counts of a hardware timer is realizable by forming an one-shot hawksbill turtle vent into a periodic hawksbill turtle vent for a long time. Hereafter, the gestalt 4 of this operation is explained using drawing 13 (A) or (C), and drawing 14. The data structure shown by drawing 1 or 5 is used for control data structure.

[0051] Drawing 13 (A) or (C) shows operation of the periodic hawksbill turtle vent in the form 4 of operation by the timing diagram. In drawing 13 (A) or (C), the cycle period of a periodic hawksbill turtle vent is set to 200ms, and operation is explained about the case where timer 1050ms is realized for a long time. As for this cycle period, it is desirable to be decided by the longest measurement range of a hardware timer (range decided by bit width of face).

[0052] Drawing 13 (A) shows operation of the first-time prolonged hawksbill turtle vent 1300. As shown in drawing 13 (A), when timer 1050ms is repeated for a long time, it decomposes into the number of times (time) repeatedly with 200ms of periods and there is a surplus, this surplus is set up as a first-time timer value 1310. In the above-mentioned example, since it is prolonged timer 1050ms=200msx5(time)+50, the first time set point of the timer value 1310 is set to 50ms from a surplus being 50. The timer operation classification 1311 is set as a long time. Like the first time timer value 1310, the periodic-start time 1312 sets the first time as 50ms, and the number of times 1314 of a repeat sets it up with 6 times (=5 time +1 time). A repeat count 1314 is set once as an excess on account of processing.

[0053] Drawing 13 (B) shows operation of the prolonged hawksbill turtle vent 1328 when there is the 1st deadline. In drawing 13 (B), the elapsed time at the time of the 1st deadline is 50ms set as the timer value 1310 at the 1st time. The remaining operation of a hawksbill turtle vent repeats the repeat period of 200ms, and should just carry out number-of-times 5 batch deadline for a long time. In this setup, since it is the part of 50ms of surpluses of drawing 13 (A), and the deadline of will already be passed once and that of will be passed 5 times 200ms of repeat periods after that, a total of six deadline (=1+5) will be performed. This is a reason for having set the repeat count 1314 as the excess once. The 1st re-connection processing removes the periodic timer queue element 1300 whose deadline was passed first, measures from the head timer queue element 1302 connected from origin next, and re-connects the periodic timer queue element 1328 to the position where a period becomes 200ms repeatedly. in the example of drawing 13 (B), since it is set to timer value (=100)=70 of the timer value (=20)-timer queue element 1306 of the timer value (=10)-timer queue element 1304 of 200ms-timer queue element 1302, 70 is set to the timer value 1330 of the timer queue element 1328. The periodic-start time 1332 is set up with 200ms, and a period counter 1334 is set as 5 times.

[0054] Drawing 13 (C) shows operation of the prolonged hawksbill turtle vent 1348 when there is the 2nd deadline. In drawing 13 (C), the periodic timer queue element 1330 whose deadline was passed first is removed, it measures from the head timer queue element 1342 connected to the degree, and the periodic timer queue element 1348 is re-connected to the position where a period becomes 200ms repeatedly. in the example of drawing 13 (C), since it is set to timer value (=10)=80 of the timer value (=30)-timer queue element 1346 of the timer value (=80)-timer queue element 1344 of 200ms-timer queue element 1342, 80 is set to the timer value 1350 of the timer queue element 1348. The periodic-start time 1352 is still 200ms, and a period counter 1354 is set as 4 times. Still, although it will be set to 4 if only 1 carries out the decrement of the period counter 1354, since 0 does not become, it is reset up in 200ms of periodic-start time like drawing 13 (C), and is reconnected to a timer queue. It repeats similarly to the 5th deadline below. Since it will be set to 0 if only 1 carries out the decrement of the period counter when there is the 6th deadline, it means completing an one-shot hawksbill turtle vent for a long time. Therefore, the re-connection with the timer queue of a periodic hawksbill turtle vent is [ that what is necessary is just to remove the timer queue element at this time from a timer queue ] unnecessary any longer. As mentioned above, the point of the form 4 of this operation is to set a surplus as the 1st time of the start as a timer value. A setup of the 2nd henceforth sets up periodic-start time. Although the values to set up differ, the setting art is the same each time.

[0055] Drawing 14 shows the flow chart of the processing in the gestalt 4 of operation. In drawing 14 , it judges whether period-counter 1332 grade is 0 (Step ST40), and when not coming out zero, it judges whether a surplus is 0 further (Step S140). Since a surplus is not 0, the first time is connected to a timer queue based on a surplus value (Step S148). After the 1st deadline, since a surplus value is 0, only 1 is subtracted from the value of a period counter (Step S144). Next, it re-connects with a timer queue based on periodic-start time 1332 grade (Step S146). Processing will be an end if a period counter is set to 0.

[0056] As mentioned above, according to the gestalt 4 of operation, the prolonged timer more than the maximum number of counts of a hardware timer is realizable by forming an one-shot hawksbill turtle vent into a periodic hawksbill turtle vent for a long time.

[0057] The gestalt 5 of gestalt 5. implementation of operation removes a queue at once using the precision of the timer value connected to a timer queue, and shows the timer management equipment and the method of integration-izing a hawksbill turtle vent which can carry out things. Hereafter, the gestalt 5 of this operation is explained using drawing 15 and drawing 16. The data structure shown by drawing 1 or 5 is used for control data structure. Since dispersion is in the precision required of the timer precision 208 by the purpose of use, with the gestalt 5 of this operation, the number of times of generating of a hawksbill turtle vent is reduced using the variation in this precision. For example, even if it removes a hawksbill turtle vent from a timer queue for 1 hour at the same time 10msec hawksbill turtle vent passes the deadline of when 10msec hawksbill turtle vent passes the deadline of, the timer precision 208 is 1 minute and the hawksbill turtle vent of 1 hour has the timer value 210 in the near, it is thought on the precision prescribe of a timer that it is satisfactory at all. In such a case, power-saving can be attained by removing from a timer queue, namely, making the deadline of pass.

[0058] Drawing 15 shows the principle of operation of the gestalt 5 of this operation. it is shown in drawing 15 -- as -- the timer queue element A1500 -- difference -- the hawksbill turtle vent on which the timer value 210 operates by 100 -- corresponding -- a degree -- difference -- the timer queue element B1510 of 2 is connected for the timer value 210. Since the timer value 210 of the timer queue element B1510 is 2 though the deadline of the timer queue element B1510 is made to pass simultaneously and an allowable error is 10 supposing the top timer queue element A1500 passes the deadline of here, it is satisfactory in any way on the precision prescribe of the timer queue element B1510. Thus, power consumption is reducible by removing simultaneously the timer queue element A1500 and the timer queue element B1510 from timer KYU. Drawing 16 shows the flow chart of the processing in the gestalt 5 of this operation. As shown in drawing 16 , it is top \*\* outside a timer queue element (Step S160) first. The following timer value and following precision of a timer queue element are measured (Step S162), and when a timer value is smaller than precision, the following timer queue element concerned is removed simultaneously (Step S164). Then, it

connects so that the timer queue element which hits the following degree may be used as the element of the head of a timer queue (Step S168). Thereby, the timer value of the timer queue element of this head is set as a hardware timer. It ends without carrying out anything at Step S162, when a timer value is larger than precision. Although not illustrated in drawing 16, the deadline of a timer queue element with the timer value of a timer queue element smaller than precision can be made to pass simultaneously by calling this timer table processing repeatedly. [0059] As mentioned above, according to the gestalt 5 of operation, since two or more timer processings can be simultaneously performed for one hawksbill turtle vent, the number of times of timer interruption can be reduced, and reduction-ization of the consumed electric current can be attained.

[0060]

[Effect of the Invention] As explained above, according to the timer management equipment and the method of this invention, the timer management equipment and the method system operation from which plurality differs with one timer equipment is realizable can be offered by managing the hawksbill turtle vent of two or more systems unitary. By treating a periodic-start hawksbill turtle vent like an one-shot type hawksbill turtle vent, management of a periodic-start hawksbill turtle vent and an one-shot type hawksbill turtle vent is unifiable, even if it is the case where there is time phase contrast further, the concurrence of a hawksbill turtle vent can be made to be able to perform, and the timer management equipment and the method of being able to reduce the number of times of timer interruption and cutting down the consumed electric current of the system by timer interruption can be offered. By forming a hawksbill turtle vent into a periodic-start hawksbill turtle vent for a long time, the timer management equipment and the method of treating a hawksbill turtle vent within the number of bits of a hardware timer for a long time can be offered. Furthermore, the timer management equipment and the method of unifying a nearby hawksbill turtle vent very much in time can be offered by using the precision of a timer value.

---

[Translation done.]

\* NOTICES \*

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

## DESCRIPTION OF DRAWINGS

---

### [Brief Description of the Drawings]

- [Drawing 1] It is drawing showing the structure of the gestalt 1 of operation, or the timer queue for every subsystem in 5.
- [Drawing 2] It is drawing showing the gestalt 1 of operation, or the data structure of the timer queue element in 5.
- [Drawing 3] It is drawing showing the data structure of the timer managed table which manages the gestalt 1 of operation, or the timer queue in 5.
- [Drawing 4] It is drawing showing the data structure of the system queue managed table which manages the gestalt 1 of operation, or the timer managed table for every system in 5.
- [Drawing 5] It is drawing showing the relation of the control structure of the gestalt 1 of operation or the system queue managed table in 5, a timer managed table, a timer queue managed table, a timer queue, etc.
- [Drawing 6] In the gestalt 1 of operation, it is the timing diagram which shows the state where timer interruption occurred at Time Ti.
- [Drawing 7] It is the flow chart which shows the processing in the gestalt 1 of operation.
- [Drawing 8] It is drawing showing operation of the periodic hawksbill turtle vent in the form 2 of operation.
- [Drawing 9] It is the timing diagram which shows operation of drawing 8.
- [Drawing 10] It is drawing showing the timer queue after reconfiguring the periodic timer queue element in the form 2 of operation.
- [Drawing 11] It is the flow chart which shows processing of the periodic hawksbill turtle vent in the form 2 of operation.
- [Drawing 12] It is the timing diagram which shows operation of the periodic hawksbill turtle vent in the form 3 of operation.
- [Drawing 13] It is the timing diagram which shows operation of the periodic hawksbill turtle vent in the gestalt 4 of operation.
- [Drawing 14] It is the flow chart of the processing in the gestalt 4 of operation.
- [Drawing 15] It is drawing showing the principle of operation of the gestalt 5 of operation.
- [Drawing 16] It is the flow chart of the processing in the gestalt 5 of operation.
- [Drawing 17] It is drawing showing the state transition of a task.
- [Drawing 18] It is drawing showing the timer managed table of the hawksbill turtle vent in realtime OS conventionally.
- [Drawing 19] It is drawing showing the example of the system which consists of two subsystems for explaining the conventional timer interruption processing.
- [Drawing 20] It is the flow chart of the conventional timer interruption processing.
- [Drawing 21] the difference currently used conventionally -- it is drawing showing the data structure of a timer queue
- [Drawing 22] the setting method of the conventional timer queue -- letting it pass -- difference -- it is drawing explaining operation of a timer
- [Drawing 23] the conventional difference -- it is the flow chart of processing using the timer queue
- [Drawing 24] It is drawing showing the managed table of the conventional periodic-start task.
- [Drawing 25] It is drawing showing operation of the task in the conventional task management method.
- [Description of Notations]
- 100, 540, 560, 580, 2100 Timer queue 102 2102 Head queue, 104, 520, 2104 Empty queue 110, 200, 535, 555, 575, 577 Timer queue element, 300, 530, 550, 570, 1920 Timer managed table, 400 510 System queue managed table 532, 552, 572 Timer queue managed table, 802 810 Periodic timer queue element 804, 806, 808, 1302, 1304, 1306, 1342, 1344, 1346, 1500, 1510 One-shot type timer queue element, 1300, 1328, 1348 Prolonged timer queue element 1700 Running state, 1710 Execution waiting state 1720 Dormant state 1850 2450 Timer table 1900 1910 System 2500 Periodic-start time 2550 2600 Task operating time.

---

[Translation done.]

Requested Patent: JP2000259429A  
Title: DEVICE AND METHOD FOR MANAGING TIMER ;  
Abstracted Patent: JP2000259429 ;  
Publication Date: 2000-09-22 ;  
Inventor(s): KAMOKI ATSUSHI ;  
Applicant(s): MITSUBISHI ELECTRIC CORP ;  
Application Number: JP19990065502 19990311 ;  
Priority Number(s): ;  
IPC Classification: G06F9/46; G04G15/00 ;  
Equivalents: ;

**ABSTRACT:**

**PROBLEM TO BE SOLVED:** To operate plural various systems with one timer device by unitarily managing the timer events of plural systems.  
**SOLUTION:** Timer queue elements 535 or the like of one (sub) system are respectively connected in the shape of queue and managed for each (sub) system by a timer queue managing table 532 or the like. The (sub) system is managed by a timer managing table 530 or the like for each (sub) system and each (sub) system is managed by a system queue managing table 510. An idle timer queue 520 successively connects used timer queues. In the case of connecting a novel timer queue element to a timer queue element 540 or the like, when there is the timer queue element 535 or the like, that novel timer queue element is connected by calculating a differential timer value but when there is no timer queue element, that novel timer queue element is connected to the head of the timer queue 540 or the like. Thus, plural timer queues are integrated, timer events are unitarily managed and plural various (sub) systems can be operated with one timer device.

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-259429

(P2000-259429A)

(43)公開日 平成12年9月22日(2000.9.22)

(51)Int.Cl.<sup>7</sup>

G 0 6 F 9/46  
G 0 4 G 15/00

識別記号

3 1 5  
3 4 0

F I

G 0 6 F 9/46  
G 0 4 G 15/00

テ-レ-ト(参考)  
3 1 5 Z 2 F 0 0 2  
3 4 0 E 5 B 0 9 8  
K

審査請求 未請求 請求項の数12 O L (全 21 頁)

(21)出願番号 特願平11-65502

(22)出願日 平成11年3月11日(1999.3.11)

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 鶴木 厚志

東京都千代田区丸の内二丁目2番3号 三菱電機株式会社内

(74)代理人 100082175

弁理士 高田 守 (外1名)

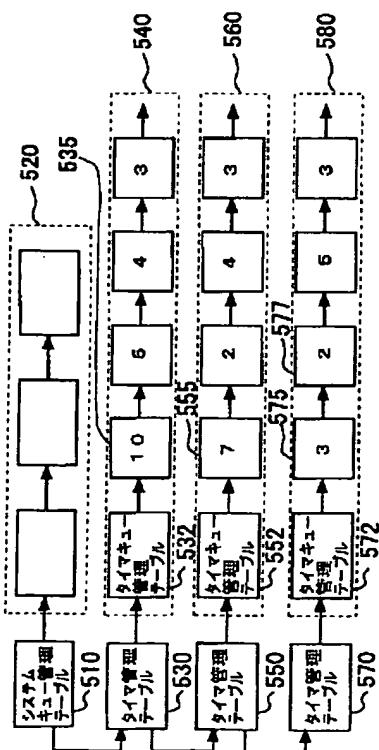
F ターム(参考) 2F002 AA04 AA07 AB04 GC07 GC22  
5B098 AA10 BA04 BB15 BB16 FF04  
CA04 GB15 GB16 GC05

(54)【発明の名称】 タイマー管理装置および方法

(57)【要約】

【課題】 一つのタイマー装置で複数の異なるシステム動作を実現し、タイマ割込みの回数を減らしてシステムの消費電流を減らすことができるタイマ管理装置および方法を提供する。

【解決手段】 複数システムのタイマイベントを一元的に管理することにより、一つのタイマー装置で複数の異なるシステム動作を実現する。周期タイマイベントをワンショット型タイマイベントと同様に扱うことにより、両タイマイベントの管理を一元化する。時間的位相差がある場合であってもタイマイベントの同時発生を行わせることにより、タイマ割込みの回数を減らすことができる。長時間タイマイベントを周期起動タイマイベント化することにより、長時間タイマイベントを扱うことができる。さらにタイマ値の精度を利用することにより、時間的にごく近傍のタイマイベントを統合することができる。



520:空きタイマキューブル  
530:650、670:タイマ管理テーブル  
531:666、676、577:タイマキューブル  
550:680、680:タイマキューブル  
551:660、680:タイマキューブル  
570:640、660:タイマキューブル  
571:640、660:タイマキューブル

## 【特許請求の範囲】

【請求項1】 タイマイベント待ち行列により管理するタイマ管理装置であって、該タイマ管理装置は、タイマイベントに対応するタイマキュー要素をタイマイベントの発行元別に待ち行列により保持する1つ以上のタイマキュー手段であって、該タイマキュー要素はタイマイベントのタイムアップ時刻と該タイマキューに接続された時刻との差分をタイマ値として有するものと、タイマ割込みが発生した場合、1つ以上の前記タイマキュー手段について、別個のタイマキュー手段の先頭に保持された別個のタイマキュー要素が有するタイマ値の中で最小のタイマ値を有する最小タイマキュー要素を選択する選択手段と、

前記最小タイマキュー要素を保持するタイマキュー手段について、該最小タイマキュー要素の次に保持されたタイマキュー要素を先頭に保持し、該最小タイマキュー要素を該タイマキュー手段から削除する並べ替え手段と、前記最小タイマキュー要素が保持されたタイマキュー手段と異なる他のタイマキュー手段について、各タイマキュー手段の先頭に保持された各タイマキュー要素が有するタイマ値から前記最小のタイマ値を減じた値を新たにタイマ値として再設定する再設定手段とを備えたことを特徴とするタイマ管理装置。

【請求項2】 前記別個のタイマキュー手段に各々対応する別個のタイマキュー管理テーブル手段と、

前記別個のタイマキュー管理テーブル手段を所定の順に管理するシステムキュー管理テーブル手段とをさらに備え、前記選択手段は、該所定の順に別個の前記タイマキュー管理テーブル手段を選択し、選択された別個の前記タイマキュー管理テーブル手段に対応する別個の前記タイマキュー手段について最小タイマキュー要素を選択することを特徴とする請求項1記載のタイマ管理装置。

【請求項3】 前記システムキュー管理テーブル手段は、前記タイマキュー管理テーブルに対応するタイマイベントの発行元の動作状態が変化した場合、前記タイマキュー管理テーブルとは異なる他のタイマキュー管理テーブルを独立して管理することを特徴とする請求項2記載のタイマ管理装置。

【請求項4】 前記システムキュー管理テーブル手段は、前記タイマキュー管理テーブルに対応するタイマイベントの発行元の存在状態が変化した場合、前記タイマキュー管理テーブルとは異なる他のタイマキュー管理テーブルを独立して管理することを特徴とする請求項2記載のタイマ管理装置。

【請求項5】 前記並べ替え手段は、前記タイマキュー手段から削除された前記最小タイマキュー要素が所定の周期時間毎に起動される周期的タイマイベントに対応する場合、削除された前記最小タイマキュー要素を前記タイマキュー手段中の該所定の周期時間後に対応する位置に再接続することを特徴とする請求項1記載のタイマ管

理装置。

【請求項6】 タイマイベントに対応するタイマキュー要素を新たに前記タイマキュー手段へ接続する接続手段をさらに備え、該接続手段は、第1周期時間毎に起動される周期的タイマイベントに対応した第1タイマキュー要素が既に前記タイマキュー手段に接続されている状態で、第1周期時間と所定の関係を有する第2周期時間毎に起動される周期的タイマイベントに対応した第2タイマキュー要素を新たに前記タイマキュー手段に接続する場合、第1タイマキュー要素の最も早いタイムアップ時刻と第2タイマキュー要素の第1回目のタイムアップ時刻を同時刻にする接続をし、1回のタイマ割込みで第1タイマキュー要素および第2タイマキュー要素を同時にタイムアップさせる回数を増やし、タイマ割込みによる消費電流を減らすことを特徴とする請求項1記載のタイマ管理装置。

【請求項7】 第2周期時間の有する第1周期時間との前記所定の関係は第2周期時間が第1周期時間の倍数である関係であり、前記接続手段は、第1タイマキュー要素がタイムアップする最も早い時刻から現時刻を引いた値を第2タイマキュー要素の第1回目のタイマ値として接続し、第1タイマキュー要素のタイムアップ時刻と第2タイマキュー要素のタイムアップ時刻を同時刻にすることを特徴とする請求項6記載のタイマ管理装置。

【請求項8】 第2周期時間の有する第1周期時間との前記所定の関係は第2周期時間が第1周期時間の約数である関係であり、前記接続手段は、第1タイマキュー要素がタイムアップする最も早い時刻から第2周期時間の整数倍を引いた時刻が現時刻より遅くならない最も早い時刻に第2タイマキュー要素を接続することを特徴とする請求項6記載のタイマ管理装置。

【請求項9】 第2周期時間の有する第1周期時間との前記所定の関係は第2周期時間が第1周期時間と互いに素である関係であり、前記接続手段は、第1周期時間と第2周期時間との最小公倍数となる周期時間毎に第1タイマキュー要素と第2タイマキュー要素とが同時にタイムアップするように接続することを特徴とする請求項6記載のタイマ管理装置。

【請求項10】 所定の起動時間を有するタイマイベントに対応するタイマキュー要素を新たに前記タイマキュー手段へ接続する接続手段をさらに備え、該接続手段は、前記タイマキュー要素を前記所定の起動時間に基づく周期時間と繰り返し回数を有し、該周期時間毎に該繰り返し回数分起動される周期的タイマイベントに対応する周期的タイマキュー要素を交換して接続することを特徴とする請求項1記載のタイマ管理装置。

【請求項11】 前記タイマキュー要素は前記タイマ値に対する許容誤差をさらに備え、前記並べ替え手段は、前記最小タイマキュー要素を保持するタイマキュー手段について、該最小タイマキュー要

素の次に保持されたタイマキュー要素の有する前記許容誤差が前記タイマ値から前記最小のタイマ値を減じた値より小さい場合、該次に保持されたタイマキュー要素を該最小タイマキュー要素と共に該タイマキュー手段から削除することを特徴とする請求項1記載のタイマ管理装置。

【請求項12】 タイマイベントを持ち行列により管理するタイマ管理方法であつて、該タイマ管理方法は、タイマイベントに対応し、該タイマイベントのタイムアップ時間に応じたタイマ値を有するタイマキュー要素を該タイマイベントの発行元別に待ち行列により1つ以上のタイマキュー手段に保持するステップと、タイマイベントに応じたタイマ割込みが発生した場合、1つ以上の前記タイマキュー手段について、別個のタイマキュー手段の先頭に保持された別個のタイマキュー要素が有するタイマ値の中で最小のタイマ値を有する最小タイマキュー要素を選択する選択ステップと、前記最小タイマキュー要素を保持するタイマキュー手段について、該最小タイマキュー要素の次に保持されたタイマキュー要素を先頭に保持し、該最小タイマキュー要素を該タイマキュー手段から削除する並べ替えステップと、前記最小タイマキュー要素が保持されたタイマキュー手段と異なる他のタイマキュー手段について、各タイマキュー手段の先頭に保持された各タイマキュー要素が有するタイマ値から前記最小のタイマ値を減じた値を新たにタイマ値として再設定する再設定ステップとを備えたことを特徴とするタイマ管理方法。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】 本発明は、タイマー管理装置および方法、特に通信、計測・制御システムなどにおいて、1つのタイマによって複数の異なるシステムのタイマ制御を行つてタイマ資源を効率的に使用するタイマ管理装置および方法に関する。

##### 【0002】

【従来の技術】 従来、通信、計測・制御等でスケジューリングなどのシステム制御を行う場合、またはリアルタイムオペレーティングシステム(OS)でタスク管理を行う場合等において使用されていたタイマ管理方法には、主として3つの方法があった。以下この3つの方法を順に説明する。

##### 【0003】 (1) イベントタイマー管理テーブルを使う方法。

1つのシステムは、機能により分割された複数の処理単位(タスク)から構成されており、1つのタスクは複数の動作状態を有している。

【0004】 図17は、タスクの状態遷移を示す。図17に示されるように、1つのタスクは複数の動作状態を持っており、この動作状態には実行状態(RUN)17

00、実行待ち状態(READY)1710、休眠状態(WAIT)1720の3つの状態がある。各タスクはその動作状況に応じて、実行状態1700と休眠状態との間(1725、1735)、休眠状態1720と実行待ち状態1710との間(1745、1755)および実行待ち状態1710と実行状態との間(1715、1705)で、その動作状態を変化(遷移)させる。サブシステムにおいて、実行状態(RUN)1700のタスク数は必ず1である。割込み等のイベントが発生すると、実行状態1700または実行待ち状態1710にある複数のタスクの内、最も優先順位の高いタスクがOSによって選択(ディスパッチ)されて実行状態へ遷移する(1715)。実行状態1700のタスクは、ディスパッチされずに実行権を他のタスクに奪われた場合は自らイベント待ちをしたい場合等には、休眠状態1720に遷移する。休眠状態1720のタスクは、イベントの発生または実行状態にあるタスクからの起床指示を受けた場合に実行待ち状態または実行状態へ遷移する(1745、1735)。

【0005】 上述のタスクに関する識別情報を管理する従来の方法を以下に説明する。図18はタスクをタイマ使用者と考えた場合の、従来のリアルタイムOSにおけるタイマイベントのタイマー管理テーブルを示す。図18(A)において、タイマ管理テーブル1850の中のタイマ使用者(タスク)の識別情報1800には、タスクID(タスク番号)、タスクの実行優先順位、タスク毎のスリックポイント等が記録されている。タイマ値1810は、動作中のタイマの残動作時間を表している。タイマ値1810が0より大きいタスクはタイマイベント待ちであり、タスクの動作状態1820は休眠状態(WAIT)である。タイマ値1810が0より大きいタスクがCPUの実行権を獲得することはない。タイマ値1810が0のタスクの動作状態1820は、実行状態(RUN)1700または実行待ち状態(READY)1710である。タスクは、システムコールを発して休眠状態1720に入るが、1つのタスクが同時に2つ以上のタイマイベントを待つて休眠状態1720に入ることは無いものとする。すなわちタスクが休眠状態1720から脱するように起動をかけるタイマイベントは1つのみであり、したがって、タイマイベントはタスク毎に1つ存在するものとする。タイマイベント(タイマ割込み)は一定時間毎に発生するものとする。この一定時間とは、多くのOSの場合約10ms程度である。タイマイベントを発行し登録した発行元タスクまたはタイマイベントが発生した時にその通知を受ける通知先タスクへ、タイマイベントの発生を何らかの信号により通知する機能があるOSの場合、タスクはタイマイベントの発生を待つて休眠状態1720へ遷移する必要はない。このようなOSの場合、タスクは休眠状態1720へ遷移せずに複数のタイマイベントの発生を同時に待つことができる。

る。一般的にはタイマイベントの発生（以下、「タイムアップ」という）の通知はメールボックスで行われる。複数のタイマイベントの発生を待つ例としては、1つ目のタイマイベントの発生により装置の電源をオンにし、2つ目のタイマイベントの発生により装置の電源をオフにする場合、または1つ目のタイマイベントAが発生する前にある処理を終了できれば正常終了としてタイマイベントAを削除するが、タイマイベントAが発生してしまうと異常であるとして停止処理を開始し、2つ目のタイマイベントBが発生してしまうと停止処理も失敗したものとして緊急停止させる等がある。

【0006】図18(B)は、上述のようにタスクが複数のタイマイベントの発生を待つ場合のタイマ管理テーブルを示す。図18(B)に示されるように、タイマ管理テーブル1800はセル1830ないしセル1835をキュー状に接続した構造（以下「タイマキュー」という）となり、図18(A)のタイマ値1810（図18(B)の符号1825）の部分にタイナキューへの先頭アドレス（ポインタ）が置かれている。このように実際のタイマはタイマキューに接続されている。

【0007】図19は、従来のタイマ割込み処理を説明するためのシステム1（1900）およびシステム2（1910）の2つのサブシステムから構成されるシステムの例を示し、図20は従来のタイマ割込み処理のフローチャートを示す。以下では図19および図20を用いて従来のタイマ割込みの処理を説明する。図19に示されるように、タイマ管理テーブル1920には各システム1900、1910毎にタイマー使用者（タスク）の識別情報（タスクID）1800およびタイマー値1810等が記録されている。各システム1900、1910は、タイマ管理テーブル1920へタイマイベント登録1905を行い、各システム1900、1910へタイムアップ1915が通知される。タイマー装置1930からの割込み1925（通常は10ms程度の周期的割り込み）によって、タイマーテーブル1920のタイマ値1810が処理される。タイマ割り込み1925が発生すると、タイマ値1810が0より大きいすべてのタスクのタイマ値1810を1減らす（デクリメントする）。初めからタイマ値1810がゼロのタスクに対しては何もしない。この処理は、登録されている全タスクについて行われる（図20のステップS202ないしS208）。タイマ割込み1925の処理において、タイマー値1810を1だけデクリメントして0になったタスク、すなわちタイマ値1810が1→0に変化したタスクに対し、タスクの状態をタイマイベント待ち状態（WAIT）1720から実行待ち状態（READY）1710に遷移させる（ステップS208、S209）。タイマ割り込み1925において、タイマ値1810のデクリメント処理（ステップS206）がすべて終了と判断されると（ステップS202）、リアルタイ

ムOS等のディスパッチ処理へ戻る（ステップS203）。ディスパッチ処理ではOSの処理の一部として、タイマ割り込み1925の処理を完了した時点で最も実行優先順位の高いタスクにCPUの実行権をわたす。具体的には、タイマ使用者（タスク）の識別情報1800の中のスタックポイントタ情報（タスクのコンテキスト（context:実行文脈））を復元して、すなわちCPUのスタックポインタレジスタに設定して、最も実行優先順位の高いタスクに実行を移す。

【0008】図20を用いて上述のタスク割り込み処理をより詳細に説明する。タイマーイベント（タイマー割込）が発生するとタイマーテーブル1850または1920の先頭から以下の動作を行う。初期設定（ステップS200）および全タイマイベントの終了判断（ステップS202）後、タイマーが使用中かどうか判定する（ステップS204）。タイマーが使用中の場合は、タイマーテーブル1850等におけるタイマ値1810が1以上の場合である。タイマーが使用中の場合は、タイマ値1810をマイナス1する（ステップS206）。タイマ値1810が0になった場合はタイムアップ処理を行う（ステップS208、S209）。タイムアップ処理S209は上述のようにタイマイベントを登録したタスクへタイマイベントの発生を通知する処理である。タイマイベントの通知方式としては上述のようにメールボックスを用いたり、タイマーテーブル1850等の中にイベントフラグを立ててOSに処理せたりする方法等がある。全タイマイベントの処理が終了するとOSのディスパッチャにジャンプする。

【0009】上述のように、従来の方法においても1個のタイマーにより複数のタスクを管理することは可能である。しかし、従来の方法では複数のサブシステムからなるシステムに対しては、タイマーテーブル1920上に複数のサブシステムが混在することになり、複数のシステム間の管理方法が非常に複雑なものとなるという問題があった。すなわち、一つのタイマ管理テーブル中に異なるシステムが混在することになり、あるシステムの運転状態が別の他のシステムのタイマの管理に影響を与えることがあった。さらにある1つのシステムが実行を中断した場合、中断したシステムも稼働中のシステムも同一のタイマーテーブル1920上に混在することになってしまうこととなる。動作を中断しているシステムのタイマをタイムアップさせることはできないため、タイマーテーブル1920におけるデクリメント処理S206は稼働中のシステムのタスクに対してだけ行い、停止中または中断中のシステムに対応するタイマーテーブル1920は変化させないという処理を行わなければならない。このように従来の方法においては、タイマ割込みを複数のシステム毎に別々に用意していたため、複数のシステム間の管理方法が非常に複雑なものとなるという問題があった。

【0010】(2) タイマキュー（差分タイマキュー）を使用する方法。

図21は従来使用されている差分タイマキューのデータ構造を示す。図21に示されるように、各システムのタイマイベント1(2118)等、タイマイベント1(2118)等の発生時間間隔のタイマ値2016等および次のシステムのタイマイベントへのポインタ（チェーン情報）2114等を1つのセル2110等として、キュー構造体（タイマキュー）2100を形成している。例えば、セル2110はシステム1のセルであり、次にシステム2のセル2120が接続されている。先頭セル2110は先頭キュー2102に接続されており、空きキュー2104には空きセル（不図示）の先頭セルが接続されている。

【0011】図22はタイマキュー2100の設定方法を通して差分タイマの動作を示す。図22において、横軸は時間軸であり、全イベント発生時間の基準時間を時間軸上で0とする。縦軸は4つの時間待ちタイマイベント(70ms, 100ms, 150ms, 200ms)を示す。これらの4つのタイマイベントは同時に動作を開始させたものとする。ハードウェアの時間計測単位（時間分解能）は、10msとする。図22のい示される4つのタイマイベント1ないし4は、設定時間の小さいものから順にタイムアップする。例えば、タイマイベント1の70msタイマイベントよりタイマイベント4の200msタイマイベントが先にタイムアップするこ

とはない。4つのタイマイベント1ないし4は、時間軸上で、70ms, 30ms, 50ms, 50msの間隔で発生する。すなわち各々の発生時刻はTE1(70ms), TE2(100ms), TE3(150ms)およびTE4(200ms)の時刻である。したがってタイマキュー2100の先頭には、イベント1(2118)のタイマ値2116が70msのセル2110を接続する。2番目に接続するセル2120のタイマ値2126には、タイマイベント2(2128)とタイマイベント1(2118)の発生時刻の差分であるTE2-TE1=100-70=30msを設定する。この差分値のことを差分タイマ値といいう。3番目以降も同様である。こうすることにより、一番目のセル2110がタイムアップしたときには70ms経過し、そこから2番目のセル2120がタイムアップするとさらに30ms経過するので、時間ゼロ（タイマ発行時）から2番目のセル2120がタイムアップするまで、 $70 + 30 = 100$ msだけ経過したことになる。同様に3番目のセル2130、4番目のセル2140も、1つ前のセルのタイムアップ時刻からの差分をタイマ値2126等として設定すれば所要のタイマ動作を実現することができる。

【0012】以下の表1に、図22における数値の計算方法を示す。

【0013】

【表1】

タイマイベント	タイムアップ時間 (ms)	タイマ設定値	差分値
タイマイベント1	70ms	7	70-0
タイマイベント2	100ms	3	100-70
タイマイベント3	150ms	5	150-100
タイマイベント4	200ms	5	200-150

【0014】表1において、タイムアップ時間はイベント毎のタイムアップ時間（開始時刻を0とする）を示す。設定値はタイマキューに設定されるタイマ値を示し、差分値は相対時間を示す。

【0015】図23は従来の差分タイマキューを用いた処理のフローチャートを示す。図23において、タイマキュー2100に有効なタイマイベントが1つ以上つながっているかどうかチェックする（ステップS230）。ここで「有効な」とはタイマキューの要素である各セル（以下「タイマキュー要素」という）のタイマ値が0より大きい（タイマ値>0）ことをいう。有効なタイマキュー要素が一つも接続されていない場合、処理を終了する。そうでない場合、タイマイベントを待っている持っているタスクに対して信号を送る（ステップS232）。タイムアップした先頭のタイマキュー要素をタイマキューから外し、その後に他のタイマキュー要素か接続されていれば、そのタイマキュー要素をタイマキュー

一管理テーブルの先頭に登録する（ステップS236）。タイマキュー先頭のタイマキュー要素のタイマ値をハードウェアタイマに設定する（ステップS238）。

【0016】上述のように、従来の方法においても1個のタイマーにより複数のタスクを管理することは可能である。しかし、従来の方法では複数のサブシステムからなるシステムに対しては、一つのタイマキュー中に異なるシステムが混在することになり、あるシステムの運転状態が別の他のシステムのタイマの管理に影響を与えることがあった。ある1つのシステムが実行を中断した場合、中断したシステムも稼働中のシステムも同一のタイマキュー中に混在することになってしまふこととなるので、あるシステムが停止中であれば、必ずしも先頭キューがタイムアップしなくなる。したがって稼働中のシステムを探索する必要が生じた。動作を中断しているシステムのタイマをタイムアップさせることはできなかった

め、タイマキューにおけるキューからの取り外し処理は稼働中のシステムのタスクに対してだけ行い、停止中または中断中のシステムに対応するタイマイベントは変化させないという処理を行わなければならないという問題があった。したがって従来の方法においては、タイマ（割込み）を複数のシステム毎に別々に用意等していたため、複数のシステム間の管理方法が非常に複雑なものとなるという問題があった。さらに、サブシステムごと休止してしまった場合、例えば2システム運転から1システム運転へと変化した場合、タイマキュー中に無効なデータ（休止中のシステムに対応するデータ）が接続していることになるため、タイマキューから外す処理（デキュー操作）が複雑になる。あるタイマキュー要素の直後のチェーン情報だけではタイマキューからデキューする操作ができない、結果として何らかの方法によりチェーン情報をたどって該当する休止システムのタイマキュー要素を探さなくてはいけないという問題があった。

#### 【0017】(3) 周期起動イベントに対するタスク管理テーブル。

定期監視動作、例えば一定時間毎の状態確認動作のように一定時間毎に動作、起動を行いたいタイマイベントに対する従来の管理方法について説明する。

【0018】図24は従来の周期起動タスクの管理テーブル2450を示す。図24において、タイマ使用者（タスク）の識別情報2400には周期起動タスクの初期設定時に周期起動時間等を記録しておく。タイマ値2410は、残動作タイマ値である。タイマ値2410が0のタスクは動作中ではないタスクを示す。タイマ値2410は、定時タイマ割込み（10ms割込み）が入る毎にマイナス1される。タイマ値2410が0になったことを検出され瞬間に、タスクはタイマイベントの発生を通知される。タイマ値2410にはタイマ使用者（タスク）の識別情報2400の周期起動時間を再設定される。タイマイベントの発生を通知されたタスクは、タスク動作を実行し終了すれば、実行待ち状態1710に戻る。すなわち実行権を放棄してディスペッチャにジャンプする。

【0019】図25(A)、(B)は従来のタスク管理方法におけるタスクの動作を示す。周期起動タスクは複数存在しうるがお互いに非同期である。図25(A)に示されるように、TE10、TE20は周期起動タスクの起動時刻であり、周期起動時間2500は本例では、 $TE20 - TE10 = 15 - 5 = 10\text{ ms}$ であり、タスク動作時間2550は $TE15 - TE10 = 10 - 5 = 5\text{ ms}$ である。このように周期起動時間2500>タスク動作時間2550の場合には、周期起動タスクの動作中にタイマ値2410が設定されて1ずつデクリメントされていっても周期起動時間2500よりも短いタスク時間2550でタスク動作が終了するためシステムの動作には何の問題も発生しない。一方図25(B)に示さ

れるように、2度目のタスク起動時刻がTE25の時刻であり、タスク動作時間2600が $TE25 - TE10 = 20 - 5 = 15\text{ ms}$ の場合、周期起動時間2500<タスク動作時間2600であるため周期起動時刻に起動されたタスクが周期起動時間2500内に完了しない。したがってTE20の時刻においてタスクが2重に起動されることとなり、システムの動作に悪影響が出るという問題があった。

【0020】上述のように従来の方法では、周期的な動作、例えば、1s毎に動作状態を表示する等の周期起動タスクのための管理テーブル2450を、タイマイベントを管理するためのタイマテーブル1850等と別に用意する必要があった。周期起動動作は、タイマテーブル1850等とは独立した機能として独立した管理テーブル2450等により処理されていた。つまり、2種類のタイマテーブル管理方法が必要であるという問題があった。このように従来のタイマ管理方法では、周期起動タイマイベントは、周期的ではないワンショット型のタイマイベント処理とは分離して、独立して管理されていた。さらに、周期起動タイマイベント同士も独立して動作していたため、同じ周期起動を持つ周期起動イベントであっても、両者にタイマイベントの発生するタイミングの差である時間的位相差があれば、タイマイベントの同時発生（同時処理）を行わせることはできなかった。このためタイマ割込みの回数が増大する傾向にあり、タイマ割込みによるシステムの消費電流が増大する傾向があるという問題があった。例えばシステムが携帯電話システムである場合、通話中であっても通話回線はハードウェア的に保持されるため、通話中であっても何もすることが無ければCPUは休眠状態(WAIT)へ遷移し、省電力化を行っている。すなわち、タスクだけではなくCPUそのものもスリープ(WAIT)させることにより省電力化を図っている。したがってタイマ割込み等によりCPUが起動されると消費電力が増大するため、このタイマ割込みの回数が増えると、確実に消費電力が増えるという問題があった。さらに、従来のタイマ管理方法では、ハードウェア的に設定された時間計測範囲よりも長い時間を図る場合には、ハードウェア上のタイマ管理とは別に、ソフトウェアによってビットサイズの大きいカウンタを設け長時間タイマイベントに対応していた。このため、ハードウェア上の時間単位や時間計測範囲が変わると、ソフトウェアによるタイマのカウント処理も影響を受けていた。さらに、従来のタイマ管理方法では、時間計測時の許容誤差という概念がなく、単純にハードウェア的に設定されたタイマ時間の終了（タイムアップ）をもってタイマイベントを発生させていた。このため、あるタイマイベントがタイムアップしたとき、その近傍に（すぐ後に）同時にタイムアップさせても時間計測上許容誤差範囲内となるタイマイベントが接続されていたとしてもデキュー処理できなかった。す

なむち従来のタイマ管理方法では、時間的にごく近傍のイベントを統合する機能が無いという問題があった。

### 【0021】

【発明が解決しようとする課題】上述のように従来のタイマ管理方法には、複数のサブシステムからなるシステムに対して1つのタイマテーブル上に複数のサブシステムを混在させていたため、複数のシステム間の管理方法が非常に複雑なものとなるという問題があった。1個のタイマーにより複数のタスクを管理することはできるものの、一つのタイマキュー中に異なるシステムが混在することになり、あるシステムの運転状態が別の他のシステムのタイマの管理に影響を与えることがあった。さらに周期起動タスクのための管理テーブルを、タイマイベントを管理するためのタイマテーブルとは別に用意する必要があった。同じ周期起動を持つ周期起動イベントであっても、時間的位相差がある場合はタイマイベントの同時発生を行わせることができなかつたため、タイマ割込みの回数が増大する傾向にあり、タイマ割込みによるシステムの消費電流が増大する傾向があるという問題があった。さらにハードウェアタイマのビット数の制限によりソフトウェアを用いて長時間タイマイベントに対応していたが、ハードウェア上条件が変わると、ソフトウェアによる処理も影響を受けていた。時間計測時の許容誤差という概念がないため、時間的にごく近傍のタイマイベントを統合する機能が無いという問題があった。

【0022】そこで、本発明の目的は、上記問題を解決するためになされたものであり、複数システムのタイマイベントを一元的に管理することにより、一つのタイマ一装置で複数の異なるシステム動作を実現することができるタイマ管理装置および方法を提供することにある。本発明の他の目的は、周期起動タイマイベントをワンショット型タイマイベントと同様に扱うことにより、周期起動タイマイベントとワンショット型タイマイベントの管理を一元化し、さらに時間的位相差がある場合であってもタイマイベントの同時発生を行わせることができ、タイマ割込みの回数を減らしてタイマ割込みによるシステムの消費電流を減らすことができるタイマ管理装置および方法を提供することにある。本発明の他の目的は、長時間タイマイベントを周期起動タイマイベント化することにより、ハードウェアタイマのビット数内で長時間タイマイベントを扱うことができるタイマ管理装置および方法を提供することにある。さらに本発明の他の目的は、タイマ値の精度を利用することにより、時間的にごく近傍のタイマイベントを統合することができるタイマ管理装置および方法を提供することにある。

### 【0023】

【課題を解決するための手段】本発明のタイマ管理装置は、タイマイベントを待ち行列により管理するタイマ管理装置であって、該タイマ管理装置は、タイマイベントに対応するタイマキュー要素をタイマイベントの発行元

別に待ち行列により保持する1つ以上のタイマキュー手段であって、該タイマキュー要素はタイマイベントのタイムアップ時刻と該タイマキューに接続された時刻との差分をタイマ値として有するものと、タイマ割込みが発生した場合、1つ以上の前記タイマキュー手段について、別個のタイマキュー手段の先頭に保持された別個のタイマキュー要素が有するタイマ値の中で最小のタイマ値を有する最小タイマキュー要素を選択する選択手段と、前記最小タイマキュー要素を保持するタイマキュー手段について、該最小タイマキュー要素の次に保持されたタイマキュー要素を先頭に保持し、該最小タイマキュー要素を該タイマキュー手段から削除する並べ替え手段と、前記最小タイマキュー要素が保持されたタイマキュー手段と異なる他のタイマキュー手段について、各タイマキュー手段の先頭に保持された各タイマキュー要素が有するタイマ値から前記最小のタイマ値を減じた値を新たにタイマ値として再設定する再設定手段とを備えたものである。ここで、本発明のタイマ管理装置は、前記別個のタイマキュー手段に各々対応する別個のタイマキュー管理テーブル手段と、前記別個のタイマキュー管理テーブル手段を所定の順に管理するシステムキュー管理テーブル手段とをさらに備え、前記選択手段は、該所定の順に別個の前記タイマキュー管理テーブル手段を選択し、選択された別個の前記タイマキュー管理テーブル手段に対応する別個の前記タイマキュー手段について最小タイマキュー要素を選択するものである。ここで、本発明のタイマ管理装置は、前記システムキュー管理テーブル手段は、前記タイマキュー管理テーブルに対応するタイマイベントの発行元の動作状態が変化した場合、前記タイマキュー管理テーブルとは異なる他のタイマキュー管理テーブルを独立して管理するものである。ここで、本発明のタイマ管理装置は、前記システムキュー管理テーブル手段は、前記タイマキュー管理テーブルに対応するタイマイベントの発行元の存在状態が変化した場合、前記タイマキュー管理テーブルとは異なる他のタイマキュー管理テーブルを独立して管理するものである。ここで、本発明のタイマ管理装置は、前記並べ替え手段は、前記タイマキュー手段から削除された前記最小タイマキュー要素が所定の周期時間毎に起動される周期的タイマイベントに対応する場合、削除された前記最小タイマキュー要素を前記タイマキュー手段中の該所定の周期時間後に対応する位置に再接続するものである。ここで、本発明のタイマ管理装置は、タイマイベントに対応するタイマキュー要素を新たに前記タイマキュー手段へ接続する接続手段をさらに備え、該接続手段は、第1周期時間毎に起動される周期的タイマイベントに対応した第1タイマキュー要素が既に前記タイマキュー手段に接続されている状態で、第1周期時間と所定の関係を有する第2周期時間毎に起動される周期的タイマイベントに対応した第2タイマキュー要素を新たに前記タイマキュー手段

に接続する場合、第1タイマキュー要素の最も早いタイムアップ時刻と第2タイマキュー要素の第1回目のタイムアップ時刻を同時刻にする接続をし、1回のタイマ割込みで第1タイマキュー要素および第2タイマキュー要素を同時にタイムアップさせる回数を増やし、タイマ割込みによる消費電流を減らすものである。ここで、本発明のタイマ管理装置は、第2周期時間の有する第1周期時間との前記所定の関係は第2周期時間が第1周期時間の倍数である関係であり、前記接続手段は、第1タイマキュー要素がタイムアップする最も早い時刻から現時刻を引いた値を第2タイマキュー要素の第1回目のタイマ値として接続し、第1タイマキュー要素のタイムアップ時刻と第2タイマキュー要素のタイムアップ時刻を同時刻にするものである。ここで、本発明のタイマ管理装置は、第2周期時間の有する第1周期時間との前記所定の関係は第2周期時間が第1周期時間の約数である関係であり、前記接続手段は、第1タイマキュー要素がタイムアップする最も早い時刻から第2周期時間の整数倍を引いた時刻が現時刻より遅くならない最も早い時刻に第2タイマキュー要素を接続するものである。ここで、本発明のタイマ管理装置は、第2周期時間の有する第1周期時間との前記所定の関係は第2周期時間が第1周期時間と互いに素である関係であり、前記接続手段は、第1周期時間と第2周期時間との最小公倍数となる周期時間毎に第1タイマキュー要素と第2タイマキュー要素とが同時にタイムアップするように接続するものである。ここで、本発明のタイマ管理装置は、所定の起動時間を有するタイマイベントに対応するタイマキュー要素を新たに前記タイマキュー手段へ接続する接続手段をさらに備え、該接続手段は、前記タイマキュー要素を前記所定の起動時間に基づく周期時間と繰り返し回数を有し、該周期時間毎に該繰り返し回数分起動される周期的タイマイベントに対応する周期的タイマキュー要素に変換して接続するものである。ここで、本発明のタイマ管理装置は、前記タイマキュー要素は前記タイマ値に対する許容誤差をさらに備え、前記並べ替え手段は、前記最小タイマキュー要素を保持するタイマキュー手段について、該最小タイマキュー要素の次に保持されたタイマキュー要素の有する前記許容誤差が前記タイマ値から前記最小のタイマ値を減じた値より小さい場合、該次に保持されたタイマキュー要素を該最小タイマキュー要素と共に該タイマキュー手段から削除するものである。

【0024】本発明のタイマ管理方法は、タイマイベントを待ち行列により管理するタイマ管理方法であって、該タイマ管理方法は、タイマイベントに対応し、該タイマイベントのタイムアップ時間に応じたタイマ値を有するタイマキュー要素を該タイマイベントの発行元別に待ち行列により1つ以上のタイマキュー手段に保持するステップと、タイマイベントに応じたタイマ割込みが発生した場合、1つ以上の前記タイマキュー手段について、

別個のタイマキュー手段の先頭に保持された別個のタイマキュー要素が有するタイマ値の中で最小のタイマ値を有する最小タイマキュー要素を選択する選択ステップと、前記最小タイマキュー要素を保持するタイマキュー手段について、該最小タイマキュー要素の次に保持されたタイマキュー要素を先頭に保持し、該最小タイマキュー要素を該タイマキュー手段から削除する並べ替えステップと、前記最小タイマキュー要素が保持されたタイマキュー手段と異なる他のタイマキュー手段について、各タイマキュー手段の先頭に保持された各タイマキュー要素が有するタイマ値から前記最小のタイマ値を減じた値を新たにタイマ値として再設定する再設定ステップとを備えたものである。

#### 【0025】

【発明の実施の形態】以下、図面を参照して、本発明の実施の形態を詳細に説明する。図1ないし図5は本発明の実施の形態1ないし5において共通して用いられるデータ構造体を示す。図1ないし図5に示されるデータ構造体によって、1個のハードウェアタイマで複数のシステム、複数の時間精度、複数の動作、例えばワンショット動作、周期動作、長時間動作等を実現する。ここで、ワンショット動作とはタイマキューにおいて1回だけ動作するタイマイベント（以下「ワンショット型タイマイベント」という）の動作であり、周期動作とはタイマキューにおいて複数回繰り返し動作するタイマイベント（以下「周期タイマイベント」という）の動作であり、長時間動作とはタイマキューにおいてハードウェアタイマのカウント数が他のタイマイベントよりも比較的長いタイマイベント（以下「長期タイマイベント」という）の動作である。これらの動作を実現するために、複数の等分タイマキューを用いて、1つのシステム内でのタイマイベントの発生タイミングを合わせるタイマキュー内の時間的位相合わせ、および複数のシステム間でのタイマイベントの発生タイミングを合わせるタイマキュー間の時間的位相合せを行う。

【0026】図1は、サブシステム毎のタイマキューの構造を示す。図1に示されるように、サブシステムのタイマイベント1(118)等、タイマイベント1(118)等の発生時間間隔のタイマ値116等および次のタイマイベントへのポインタ（チェーン情報）114等を1つのセル（タイマキュー要素）110等として、キューモード（タイマキュー）100を形成している。例えば、タイマキュー要素110の次にタイマキュー要素120が接続されている。図1では図示されていないがタイマキューは双方向の線形リスト構造である。先頭タイマキュー要素110は先頭キー102に接続されており、空きキー104には空きセル（不図示）の先頭セルが接続されている。

【0027】図2はタイマキュー要素のデータ構造を示す。図2において、前方リンク202はタイマキュー内

の前方タイマキュー要素へのポインタ、後方リンク204はタイマキュー内の後方タイマキュー要素へのポインタ、タイマID206はタイマイベント、したがって当該タイマキュー要素の識別子、タイマ種別・精度208は動作中のハードウェアタイマのビット幅またはタイマカウント単位（分解能、精度等）、タイマ値210は設定するタイマ値である。タイマ値210は本明細書においては差分タイマ値を用いるが、本発明は差分タイマ値に限定されるものではない。さらに図2において、通知先情報212はタイマイベントを発行したオブジェクトのID、タイマ動作種別214はワンショット動作、周期動作または長時間ワンショット動作（長時間動作でかつワンショット動作）の種別、周期起動時間216は周期動作を行う周期タイマイベントの周期起動時間または長時間ワンショット動作を行う長時間タイマイベントの初期設定値（後述の実施の形態4で示されるように長時間タイマイベントは周期タイマイベント化される）、周期起動カウンタ218は周期起動タイマイベントまたは長時間タイマイベントの繰り返し回数、システム種別220は誤接続を検出するための動作中のシステム種別である。タイマキュー要素200にはさらのその他のタイマキュー要素に関する情報を設定することもできる。

【0028】タイマキューの先頭タイマキュー要素の前にはタイマキュー管理テーブルがある。このタイマキュー管理テーブルは図1における先頭キュー102が該当する。

【0029】図3はタイマキュー管理テーブルの前に接続され、タイマキューを管理するタイマ管理テーブルのデータ構造を示す。タイマ管理テーブルはシステム毎に用意される。図3において、前システムへのポインタ302は当該タイマ管理テーブル300により管理されるシステムの前のシステムへのポインタ（リンク）、次システムへのポインタ304は当該タイマ管理テーブル300により管理されるシステムの次のシステムへのポインタ（リンク）、先頭キューへのポインタ306は当該システムのタイマキュー管理テーブルへのポインタ、システムの動作状態308はシステムの動作中、停止中等のシステムの動作状態を表示するためのフラグまたはステータス、システム種別310は動作中のシステム種別である。タイマ管理テーブル300にはさらのその他の当該システムに関する情報を設定することもできる。

【0030】図4は各システム毎のタイマ管理テーブル300を管理するシステムキュー管理テーブルのデータ構造を示す。図4において、システムへのリンク情報402はシステムキュー管理テーブル400が管理する各システムをサブシステムとし、これらのサブシステムから構成されるシステムを全体として1つのシステムとした場合、他のシステムへのポインタである。さらに図4において、先頭キューへのポインタ404は先頭（サブ）システムのタイマ管理テーブル300へのポイン

タ、空きキューへのポインタ406は空きタイマキューへのポインタ、システムの動作状態408は全体として1つのシステムとした場合における当該システムの動作中、停止中または中断中等の状態、その他管理情報410は当該システムの他の情報、システム種別412は動作中の当該システム種別である。

【0031】図5は上述のシステムキュー管理テーブル400、タイマ管理テーブル300、タイマキュー管理テーブルおよびタイマキュー等の制御構造の関連を示す。図5に示されるように、1つの（サブ）システムのタイマキュー要素535等は各々キュー状に接続され、このキューはタイマキュー管理テーブル532等によって（サブ）システム毎に管理される。タイマキューという場合、このタイマ管理テーブル532等を含む全体540等を指すが、単にタイマキュー要素535から始まるキューを指す場合もある。（サブ）システムは、（サブ）システム毎のタイマ管理テーブル530等によって管理され、各（サブ）システムはシステムキュー管理テーブル510によって管理される。（サブ）システムの例としては、符号分割多重（Code Division Multiple Access : CDMA）システム、ディジタル携帯電話／自動車電話（Personal Digital Cellular : PDC）システムまたはPHS（Personal Handyphone System）等がある。空きタイマキュー520は、タイマがカウントアップして処理が終了したタイマキュー要素、つまり使い終わったタイマキュー要素が接続されていく。空きタイマキュー要素が無くなると新規のタイマイベントは発行できないため、適宜ガーベッジコレクション等を行う。新規なタイマキュー要素をタイマキュー540等に接続する場合、タイマキュー540等にタイマキュー要素535等がある場合は差分タイマ値を計算して新規なタイマキュー要素をタイマキューに接続する。一方タイマキュー要素が無い場合は、タイマキュー540等の先頭にタイマ値210をそのまま設定して新規なタイマキュー要素をそのまま接続する。

【0032】実施の形態1. 実施の形態1は、複数の（サブ）システムのタイマキューを統合し、複数の（サブ）システムのタイマイベントを一元的に管理することにより、一つのタイマー装置で複数の異なる（サブ）システムの動作を実現することができるタイマ管理装置および方法を示す。以下では、特に明示されない限りシステムといえばタイマ管理テーブル530等で管理される各システムを指す。システムキュー管理テーブル510で管理される全体として1つのシステムとの関係に及ぶ場合は、全体として1つのシステムをシステムといい、タイマ管理テーブル530等で管理される各システムをサブシステムという。実施の形態1ではシステム毎のタイマキュー間の時間的位相合わせ方法、およびタイマイベントが発生した時に、該タイマイベントに対応するシステムの先頭タイマキュー要素を外す操作を中心として

説明する。制御データ構造は、図1ないし5で示されたデータ構造等を使用する。

【0033】図6は、3つの(サブ)システム1ないし3が存在する場合において、時刻T<sub>i</sub>でタイマ割込みが発生した状態のタイムチャートを示す。各システム1ないし3は、各々図5におけるタイマキュー540ないし580を有しており、各タイマキュー540等は、各システム1ないし3毎に独立である。図6において、時刻T<sub>i</sub>でタイマ割込みが発生した場合、各システム毎のタイマキュー540等の先頭のタイマキュー要素535等を比較して最小のタイマ値210を有するタイマキュー要素を選択する。図5の例では、システム1の先頭タイマキュー要素535のタイマ値210は10、システム2の先頭タイマキュー要素555のタイマ値210は7、システム3の先頭タイマキュー要素575のタイマ値210は3であるため、システム3のタイマ値210の3が最小である。したがって、システム3の先頭タイマキュー要素575がタイムアップしたと判断できる。そこでシステム3の先頭タイマキュー要素575をタイマキュー580から外して、次のタイマキュー要素577(タイマ値210は2)をシステム3の先頭タイマキュー要素とする。外されたタイマキュー要素は空きタイマキュー520に接続される。以上のようにしてシステム3におけるタイマキュー要素575のタイムアップ時の操作が完了する。この時システム1とシステム2のタイマキュー要素535および555はタイムアップしていないが、各先頭タイマキュー要素のタイムアップ時間はシステム3の先頭タイマキュー要素575のタイマ値210である3の時間経過分だけ少なくする必要がある。そこでシステム1、2の各先頭タイマキュー要素535および555から、タイムアップしたシステム3のタイマキュー要素575のタイマ値210の3を引いた値をタイマ値210として再設定する。システム1の場合タイマ値210は10-3=7、システム2の場合タイマ値210は、7-3=4が新しいタイマ値となり、各々先頭タイマキュー要素535、555のタイマ値210の内容となる。各システム1ないし3の先頭タイマキュー要素のタイマ値210を再設定した後、その値の最小のものをハードウェアタイマに設定し、タイマテーブル処理を完了する。本例の場合、システム3のタイマ値210が2と最小であるため、ハードウェアタイマには2が設定される。本例では、1つのタイマキュー要素575だけタイムアップしたが、設定されているタイマ値210によっては、複数のシステムのタイマキュー要素が同時にタイムアップすることがある。この場合は、タイムアップしたタイマキュー要素はすべてタイマキュー540等から外す処理を行う。

【0034】図7は本実施の形態1における処理をフローチャートで示す。図7において、各システム間でタイマキュー540等の先頭タイマキュー要素535等の

値を比較する。ただし、タイマ管理テーブル530等のシステムの動作状態308により、システムが停止中または実装されていない場合、タイマキュー540等に有效的なタイマキュー要素が無い場合等で動作していないタイマキュー540等に対しては処理を行わない。この場合は、タイマ管理テーブル530等の次システムへのポインタ304をたどって次のシステムのタイマキュー560等の処理を行う。同時にシステム種別310を記録しておく(ステップS70)。タイマがタイムアップした時、先頭タイマキュー要素575等のタイマ値210が最小のタイマキュー要素575等がタイムアップしたタイマキュー要素であるので、このタイマキュー要素575等タイマキュー580等から外す。2つのシステムが同時にタイムアップする場合、すなわち先頭タイマキュー要素575等の値が同じ場合は2つのタイマキュー要素575等を外す。3つ以上同じ場合も同様に外す。上述のシステム間でのタイマ値210の比較は、タイマ管理テーブル530等の次システムへのポインタ304をたどることにより行われる。したがってシステムを構成するサブシステムの数によって本実施の形態の処理方法が影響を受けることはない。システムの動作中にあるサブシステムが停止または削除されても、次システムへのポインタ304をたどって探索される際にそのシステムがスキップされるだけであるため、他の動作中のサブシステムの処理に影響を与えることはない。次に全サブシステムの先頭タイマキュー要素535等のタイマ値210を補正する。初期設定としてシステム番号カウンタ<sub>i</sub>をゼロクリアする(ステップS72)。全サブシステムに対しての処理が終了するまでステップS76からS79の処理を繰り返す(ステップS74)。まず、サブシステムがタイムアップしたサブシステムかどうかを判定する(ステップS76)。もしタイムアップしたサブシステムならばタイマ値210の調整は不要であるので、ステップS79へとぶ。そうでない場合は、タイムアップしてタイマキュー580等から外されたタイマキュー要素575等の差分タイマ値210を、他のタイムアップしなかったサブシステムの先頭タイマキュー要素555等の差分タイマ値210から引く(ステップS78)。このデクリメントによりサブシステム間のタイマキュー540等の時間位相を合わせることができる。次のサブシステムの先頭タイマキュー要素535等のタイマ値210の調整を行うために、システム番号カウンタ<sub>i</sub>に1を加算する(ステップS79)。ステップS74へ戻り、全サブシステムについて完了した場合は、先頭タイマキュー要素555等のタイマ値210の内最小の値を探して、ハードウェアタイマに設定する。

【0035】上述の実施の形態1において、サブシステムごと削除する場合は、先頭タイマキュー要素535等をシステムキュー管理テーブル510につないで、そのタイマキュー管理テーブル532等の内容を空にするだ

けで実現できる。逆にシステムを新規に増設する場合は、独立の新規なタイマキュー、タイマキュー管理テーブルおよびタイマ管理テーブルを生成して、このタイマ管理テーブルをシステムキュー管理テーブル510の管理する他のサブシステムのタイマ管理テーブルのリストに追加し、新規なシステムの動作状態を示すシステムの動作状態308を動作中に実現できる。したがって、既存の他の部分に全く影響を与えることなく新規システムの増設を行うことができる。

【0036】以上より、実施の形態1によれば、複数のサブシステムのタイマイベントを各サブシステム毎のタイマキューを用いて一元的に管理することにより、一つのハードウェアタイマーで複数の異なるサブシステムの動作を実現することができる。

【0037】実施の形態2. 実施の形態2は、周期タイマイベントを一般的のワンショット型タイマイベントと同様に扱うことにより、周期タイマイベントとワンショット型タイマイベントの管理を一元化することができるタイマ管理装置および方法を示す。以下、本実施の形態2を図8ないし11を用いて説明する。制御データ構造は、図1ないし5で示されたデータ構造等を使用する。タイマキュー540等において、1回だけ動作するワンショット型タイマイベントと設定値により繰返し動作する周期タイマイベントとを同時に扱うために、図2のタイマキュー要素200中に動作識別情報をタイマ動作種別214、周期起動時間216として設けている。タイマ動作種別214が周期動作であるタイマキュー要素（以下「周期タイマキュー要素」という。これと区別するためにワンショット型タイマイベントに対応するタイマキュー要素を特にワンショット型タイマキュー要素という。）は、周期起動時間216毎に周期起動される。

【0038】図8は実施の形態2における周期タイマイベントの動作を示す。図8において、タイマキューの先頭の周期タイマキュー要素802（周期起動時間216=10）がタイムアップしたとする。この時、周期タイマキュー要素802は、タイマキューの先頭から外されると、周期起動時間216 (=10) 後には再び起動されるため、タイマキューの周期起動時間216 (=10) 後の位置810に再接続される。図8に示されるように、ワンショット型タイマキュー要素804、806の後に周期起動のタイマ値を3として再接続される。

【0039】図9は図8の動作をタイムチャートで示す。図9に示されるように、周期タイマキュー要素802がタイムアップする時刻をT<sub>A</sub>、ワンショットタイマキュー要素804がタイムアップする時刻をT<sub>B</sub>、ワンショットタイマキュー要素806がタイムアップする時刻をT<sub>C</sub>、ワンショットタイマキュー要素808がタイムアップする時刻をT<sub>E</sub>とすると、再接続すべき周期タイマキュー要素810はその周期起動時間216が10であるから、10-2-5=3の時間だけ経過した時刻

T<sub>D</sub>の位置に、タイマ値210を3に設定して再接続すればよいことがわかる。周期タイマキュー要素810を再設定した後のタイマキューを、図10に示す。

【0040】図11は、実施の形態2における周期タイマイベントの処理をフローチャートで示す。図11において、タイマがタイムアップした時、タイマ発行元へイベント通知を行う（ステップS110）。この処理はワンショット動作のタイマイベントと同じ処理である。次に、周期タイマイベントのタイムアップかどうかを判断し（ステップS112）、そうであればタイムアップした周期タイマイベントをタイマキューに再接続する（ステップS114）。周期タイマキュー要素の次のタイマキュー要素が無かった場合は、周期タイマイベントが再びタイマキューの先頭に接続されることになる。最後に先頭タイマキュー要素のタイマ値210をハードウェアタイマーに設定する（ステップS116）。ステップS112で周期タイマイベントのタイムアップでない場合は直接ステップS116へとぶ。

【0041】本実施の形態2によれば、同一の周期タイマイベントがタイマキュー上に2つ以上存在しないのでメモリを節約でき、かつタイマキューを探索する時間の節約を図ることができる。周期タイマイベントを停止する場合は、タイマキュー上からその周期タイマイベントを1つだけ削除すればよい。さらに、本実施の形態2により、ハードウェア的にタイマの最大カウント数（タイマのビット幅）が少ない場合でも、タイマキュー要素を周期的に使うことにより、ハードウェア上の制約を無くすことができる。例えば、タイマの分解能が10msで周期起動時間が1時間を最長すると、1時間=60(分)×60(秒/分)×100(カウント/秒)=360,000カウントであるから、この1時間の長期タイマイベントを実現するハードウェアタイマには約20ビット必要となる。一方、実施の形態2によりこの周期起動時間が1時間の長期タイマイベントを周期化して、例えば10ms分解能で1分を最長とする周期タイマキュー要素を用いると、1分=60秒×100(カウント/秒)=6,000カウントとなる。したがって、この1時間の長期タイマイベントを実現するハードウェアタイマは16ビットのビット幅で実現可能となり、約4ビット少なくて済む。

【0042】以上より、実施の形態2によれば、周期タイマイベントをワンショット型のタイマイベント化することにより、一般的のワンショット型タイマイベントと同様に扱うことができるため、周期タイマイベントとワンショット型タイマイベントの管理を一元化することができる。

【0043】実施の形態3. 実施の形態3は、周期タイマイベント間の時間的位相合わせを行うことによってハードウェアタイマのタイムアップ回数を減らすことができるタイマ管理装置および方法を示す。以下、本実施の

形態3を図12(A)、(B)を用いて説明する。制御データ構造は、図1ないし5で示されたデータ構造等を使用する。

【0044】図12(A)、(B)は実施の形態3における周期タイマイベントの動作をタイムチャートで示す。図12(A)に示すように、例えば、周期起動時間216が1秒毎の周期タイマイベント1と周期起動時間216が2秒毎の周期タイマイベント2があるものとする。この場合、時間的位相合わせができていないと、すなわちタイマイベント1とタイマイベント2が非同期であると、タイマイベント1がT2、T3、T5、T6およびT8の計5回発生し、タイマイベント2がT1、T4およびT7の計3回発生し、総計8回発生している。

【0045】一方、図12(B)に示すように2つの周期タイマイベント1およびタイマイベント2が発生する時間的位相を合わせることによって、タイミングT1、T2、T14およびT16で2つのタイマイベント1とタイマイベント2とを同時にタイムアップさせることができる。これによって、タイマイベントの発生回数を8回から5回に減らすことができる。以下、2つのタイマイベントの時間的位相を合わせる方法について説明する。図12(B)に示されるように、周期タイマイベント2を起動させる時(Tz)は、タイマキュー上すでに動いている他の周期タイマイベント1の時間的位相に合わせてタイマイベント2の発行時間を調整し、タイマキューに接続を行う。タイマイベント2の登録を行う時(Tz)、他の周期タイマイベント1のタイムアップ時間T12とTzとの差分 $\Delta t = T12 - Tz$ を計算して、この $\Delta t$ を周期タイマイベント2の初めの(1回目の)タイマ値210として設定する。すなわちタイマ値210 =  $\Delta t$ 、タイマ動作種別214 = 周期、周期起動時間216 = 周期起動したい時間等に設定する。このように他の周期タイマイベント1の発生周期(=1)に同期するように、最初の1回だけタイマイベント2のタイマ値210を本来の周期起動時間216(=2)ではなく $\Delta t$ に設定する。T12で一度同期がとれると、後はT14、T16とも同期がとれていくことになる。この結果ハードウェアタイマが発生するタイマ割り込みの回数を減らすことができる。したがってタイマ割り込み等によりCPUが起動されて消費される電力を減らすことができるため、携帯電話等の電池駆動型のシステムの場合には大きく省電力化を測ることができる。

【0046】例えば、周期起動時間216が6の周期タイマイベントに周期起動時間216が12の周期タイマイベントと72のタイマイベントが追加される場合の動作を説明する。周期起動時間216が6の周期に時間的位相を合わせて、周期起動時間216が12、起動時間216が72のタイマイベントを発行する。両タイマイベントに対応するタイマキュー要素がタイマキューに接

続される。この場合、周期起動時間216の6、12、72には最大公約数6があるので、周期起動時間216が12、72の2つのタイマイベントを追加しても、CPUがタイマイベントを処理するために起動される回数は増えない。最大公約数があれば、メモリ等の条件を除いて追加されるタイマイベントの数に制限はない。

【0047】逆に周期起動時間216が72のイベントに周期起動時間216が6のタイマイベントを接続する場合も同様である。周期起動時間216が72のイベントに周期起動時間216が6のタイマイベントを接続する場合、72と6との最大公約数は6であるので、周期起動時間216が72のタイマイベントがタイムアップする時刻tから $6 \times n$ 時間引きた数が0になる最大のnを与えるタイミングで、周期起動時間216が6のイベントを発行してタイマキューに接続すればよい。最大のnを与えるタイミングとするのは、最も早いタイミングとするためである。

【0048】周期起動時間216が互いに素な場合の例を示す。周期起動時間216が5のタイマイベントが動作中に、周期起動時間216が3のタイマイベントを発行するものとする。このとき周期起動時間216が5のイベントと周期起動時間216が3のイベントとは互いに素である、すなわち互いに素数で公約数がない。この場合は、周期起動時間216が3と5との最小公倍数15(3×5)となる時間に2つのタイマイベントの発生が重なるように、周期起動時間216が3のタイマイベントの発行タイミングを調整すればよい。

【0049】以上より、実施の形態3によれば、周期タイマイベント間の時間的位相合わせを行うことによってハードウェアタイマのタイムアップ回数を減らすことができるため、タイマ割込み等によりCPUが起動されて消費される電力を減らすことができる。

【0050】実施の形態4。実施の形態4は、長時間ワンショットタイマイベントを周期タイマイベント化することによって、ハードウェアタイマの最大カウント数以上の長時間タイマを実現することができるタイマ管理装置および方法を示す。以下、本実施の形態4を図13(A)ないし(C)、図14を用いて説明する。制御データ構造は、図1ないし5で示されたデータ構造等を使用する。

【0051】図13(A)ないし(C)は、実施の形態4における周期タイマイベントの動作をタイムチャートで示す。図13(A)ないし(C)において、周期タイマイベントの繰返し周期を200msとし、長時間タイマ1050msを実現する場合について動作を説明する。この繰返し周期は、ハードウェアタイマの最長計測範囲(ビット幅によって決まる範囲)によって決められることが望ましい。

【0052】図13(A)は初回の長時間タイマイベント1300の動作を示す。図13(A)に示されるよう

に、長時間タイマ1050msを繰り返し周期200msと繰り返し回数(回)に分解し、剩余がある場合はこの剩余を初回のタイマ値1310として設定する。上記の例では長時間タイマ1050ms = 200ms × 5(回) + 50であるため、剩余が50であることよりタイマ値1310の初回設定値は50msとなる。タイマ動作種別1311は長時間に設定する。周期起動時間1312は初回タイマ値1310と同様に初回は50msに設定し、繰り返し回数1314は6回(=5回+1回)と設定する。繰り返し回数1314は、処理の都合上1回余分に設定する。

【0053】図13(B)は1回目のタイムアップがあった場合の長時間タイマイベント1328の動作を示す。図13(B)において、1回目のタイムアップ時の経過時間は、1回目にタイマ値1310に設定した50msである。長時間タイマイベントの残りの動作は、200msの繰り返し周期を繰り返し回数5回分タイムアップすればよい。この設定では図13(A)の剩余50msの分すでに1回タイムアップしており、その後繰り返し周期200msで5回タイムアップすることになるため、計6回(=1+5)のタイムアップを行うことになる。これが繰り返し回数1314を1回余分に設定した理由である。1回目の再接続処理は、まずタイムアップした周期タイマキュー要素1300を外し、次に元から接続されていた先頭タイマキュー要素1302から起算して繰り返し周期が200msとなる位置に周期タイマキュー要素1328を再接続する。図13(B)の例では、200ms - タイマキュー要素1302のタイマ値(=10) - タイマキュー要素1304のタイマ値(=20) - タイマキュー要素1306のタイマ値(=100) = 70となるため、タイマキュー要素1328のタイマ値1330には70が設定される。周期起動時間1332は200msと設定し、周期カウンタ1334は5回に設定される。

【0054】図13(C)は2回目のタイムアップがあった場合の長時間タイマイベント1348の動作を示す。図13(C)において、まずタイムアップした周期タイマキュー要素1330を外し、次に接続されていた先頭タイマキュー要素1342から起算して繰り返し周期が200msとなる位置に周期タイマキュー要素1348を再接続する。図13(C)の例では、200ms - タイマキュー要素1342のタイマ値(=80) - タイマキュー要素1344のタイマ値(=30) - タイマキュー要素1346のタイマ値(=10) = 80となるため、タイマキュー要素1348のタイマ値1350には80が設定される。周期起動時間1352は200msのままであり、周期カウンタ1354は4回に設定される。周期カウンタ1354を1だけデクリメントすると4となるが、まだ0とはならないので、図13(C)と同様にして周期起動時間200msで設定し直してタ

イマキューに接続しなおす。以下5回目のタイムアップまで同様に繰り返す。6回目のタイムアップがあった場合は、周期カウンタを1だけデクリメントすると0になるので、長時間ワンショットタイマイベントは完了したことになる。したがってこのときのタイマキュー要素をタイマキューから外せばよく、周期タイマイベントのタイマキューへの再接続はもう必要ない。上述のように、本実施の形態4のポイントは、初めの1回目に剩余をタイマ値として設定することにある。2回目以降の設定は周期起動時間を設定する。設定する値が異なるが設定処理方法は毎回同じである。

【0055】図14は実施の形態4における処理のフローチャートを示す。図14において、周期カウンタ1332等が0であるか否かを判断し(ステップS140)、0出ない場合はさらに剩余が0であるか否かを判断する(ステップS140)。初回は剩余が0ではないため剩余値を元にタイマキューに接続する(ステップS148)。1回目のタイムアップ以後は剩余値が0であるため、周期カウンタの値から1だけ引く(ステップS144)。次に周期起動時間1332等を元にタイマキューに再接続する(ステップS146)。周期カウンタが0になれば処理は終了である。

【0056】以上より、実施の形態4によれば、長時間ワンショットタイマイベントを周期タイマイベント化することによって、ハードウェアタイマの最大カウント数以上の長時間タイマを実現することができる。

【0057】実施の形態5・実施の形態5は、タイマキューに接続されるタイマ値の精度を利用して一度にキューを外し、タイマイベントを統合化することできるタイマ管理装置および方法を示す。以下、本実施の形態5を図15および図16を用いて説明する。制御データ構造は、図1ないし5で示されたデータ構造等を使用する。タイマ精度208には使用目的によって要求される精度にばらつきがあるため、本実施の形態5では、この精度のバラツキを利用してタイマイベントの発生回数を減らすものである。例えば、10mssecタイマイベントがタイマアップした時、その近傍にタイマ精度208が1分であり、かつタイマ値210が1時間のタイマイベントがあった場合、10mssecタイマイベントがタイマアップすると同時に1時間タイマイベントをタイマキューから外しても、タイマの要求精度上、何ら問題ないと考えられる。このような場合、タイマキューから外す、すなわちタイマアップさせることによって、省電力化を図ることができる。

【0058】図15は本実施の形態5の動作原理を示す。図15に示されるように、タイマキュー要素A1500は差分タイマ値210が100で動作するタイマイベントに対応し、次に差分タイマ値210が2のタイマキュー要素B1510が接続されている。ここで先頭のタイマキュー要素A1500がタイマアップしたとする

と、タイマキュー要素B1510を同時にタイムアップさせたとしても、タイマキュー要素B1510のタイマ値210は2であり許容誤差は10であるため、タイマキュー要素B1510の要求精度上何ら問題はない。このようにタイマキュー要素A1500とタイマキュー要素B1510とを同時にタイマキューから外すことによって消費電力を削減することができる。図16は本実施の形態5における処理のフローチャートを示す。図16に示されるように、まず先頭のタイマキュー要素外す(ステップS160)。次のタイマキュー要素のタイマ値と精度とを比較して(ステップS162)、タイマ値が精度より小さい場合は、当該次のタイマキュー要素を同時に外す(ステップS164)。この後、次の次にあたるタイマキュー要素をタイマキューの先頭の要素とするように接続する(ステップS168)。これによりこの先頭のタイマキュー要素のタイマ値がハードウェアタイマに設定される。ステップS162でタイマ値が精度より大きい場合は何もしないで終了する。図16では図示されていないが、本タイマテーブル処理が繰り返し呼ばれることにより、タイマキュー要素のタイマ値が精度より小さいタイマキュー要素を同時にタイムアップさせていくことができる。

**[0059]** 以上より、実施の形態5によれば、1つのタイマイベントで同時に2つ以上のタイマ処理をおこなうことができるため、タイマ割込み回数を減らすことができ、消費電流の低減化を達成することができる。

#### **[0060]**

**【発明の効果】** 以上説明したように、本発明のタイマ管理装置および方法によれば、複数システムのタイマイベントを一元的に管理することにより、一つのタイマー装置で複数の異なるシステム動作を実現することができるタイマ管理装置および方法を提供することができる。周期起動タイマイベントをワンショット型タイマイベントと同様に扱うことにより、周期起動タイマイベントとワンショット型タイマイベントの管理を一元化し、さらに時間的位相差がある場合であってもタイマイベントの同時発生を行わせることができ、タイマ割込みの回数を減らしてタイマ割込みによるシステムの消費電流を減らすことができるタイマ管理装置および方法を提供することができる。長時間タイマイベントを周期起動タイマイベント化することにより、ハードウェアタイマのビット数内で長時間タイマイベントを扱うことができるタイマ管理装置および方法を提供することができる。さらに、タイマ値の精度を利用することにより、時間的にごく近傍のタイマイベントを統合することができるタイマ管理装置および方法を提供することができる。

#### **【図面の簡単な説明】**

**【図1】** 実施の形態1ないし5におけるサブシステム毎のタイマキューの構造を示す図である。

**【図2】** 実施の形態1ないし5におけるタイマキュー

要素のデータ構造を示す図である。

**【図3】** 実施の形態1ないし5におけるタイマキューを管理するタイマ管理テーブルのデータ構造を示す図である。

**【図4】** 実施の形態1ないし5における各システム毎のタイマ管理テーブルを管理するシステムキュー管理テーブルのデータ構造を示す図である。

**【図5】** 実施の形態1ないし5におけるシステムキュー管理テーブル、タイマ管理テーブル、タイマキュー管理テーブルおよびタイマキュー等の制御構造の関連を示す図である。

**【図6】** 実施の形態1において、時刻Tiでタイマ割込みが発生した状態を示すタイムチャートである。

**【図7】** 実施の形態1における処理を示すフローチャートである。

**【図8】** 実施の形態2における周期タイマイベントの動作を示す図である。

**【図9】** 図8の動作を示すタイムチャートである。

**【図10】** 実施の形態2における周期タイマキュー要素を再設定した後のタイマキューを示す図である。

**【図11】** 実施の形態2における周期タイマイベントの処理を示すフローチャートである。

**【図12】** 実施の形態3における周期タイマイベントの動作を示すタイムチャートである。

**【図13】** 実施の形態4における周期タイマイベントの動作を示すタイムチャートである。

**【図14】** 実施の形態4における処理のフローチャートである。

**【図15】** 実施の形態5の動作原理を示す図である。

**【図16】** 実施の形態5における処理のフローチャートである。

**【図17】** タスクの状態遷移を示す図である。

**【図18】** 従来リアルタイムOSにおけるタイマイベントのタイマ管理テーブルを示す図である。

**【図19】** 従来のタイマ割込み処理を説明するための2つのサブシステムから構成されるシステムの例を示す図である。

**【図20】** 従来のタイマ割込み処理のフローチャートである。

**【図21】** 従来使用されている差分タイマキューのデータ構造を示す図である。

**【図22】** 従来のタイマキューの設定方法を通して差分タイマの動作を説明する図である。

**【図23】** 従来の差分タイマキューを用いた処理のフローチャートである。

**【図24】** 従来の周期起動タスクの管理テーブルを示す図である。

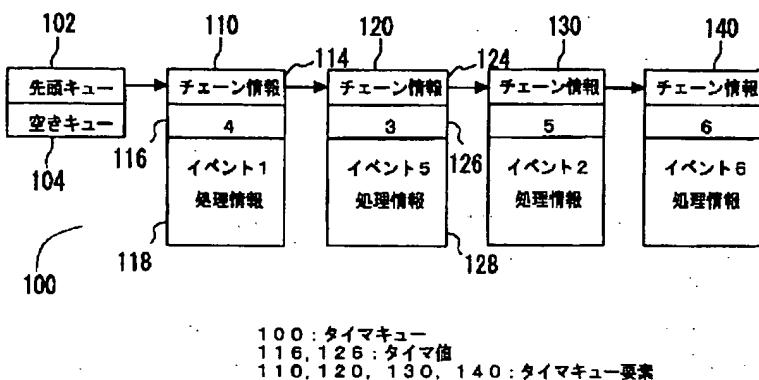
**【図25】** 従来のタスク管理方法におけるタスクの動作を示す図である。

**【符号の説明】**

100、540、560、580、2100 タイマキュー、102、2102 先頭キュー、104、520、2104 空きキュー、110、200、535、555、575、577 タイマキュー要素、300、530、550、570、1920 タイマ管理テーブル、400、510 システムキュー管理テーブル、532、552、572 タイマキュー管理テーブル、802、810 周期タイマキュー要素、

804、806、808、1302、1304、1306、1342、1344、1346、1500、1510 ワンショット型タイマキュー要素、1300、1328、1348 長時間タイマキュー要素、1700 実行状態、1710 実行待ち状態、1720 休眠状態、1850、2450 タイマテーブル、1900、1910 システム、2500 周期起動時間、2550、2600 タスク動作時間。

【図1】



【図2】

前方リンク	202
後方リンク	204
タイマID (タイマ別子)	206
タイマ種別・頻度 (タイマ単位: 分解能)	208
タイマ値 (送分ターゲット値、ターゲット容限差)	210
通知先情報 (通知先ID、通知方法など)	212
タイマ動作種別 (ワンショット/周期/長時間ワンショット)	214
周期起動時間 (周期/長時間ワンショットの初期設定値)	216
周期起動カウント (周期/長時間ワンショットの周回カウント)	218
システム種別 (接続接続用)	220

200: タイマキュー要素

【図3】

前システムへのポインタ	302
次システムへのポインタ	304
先頭キューへのポインタ	306
システムの動作状態 (動作中/停止中/未実装等)	308
システム種別	310

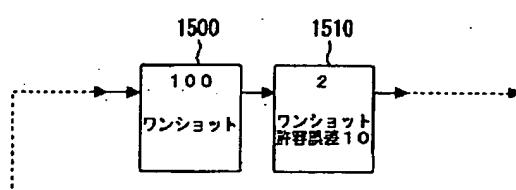
300: タイマ管理テーブル

【図4】

システムへのリンク情報	402
先頭キューへのポインタ	404
空きキューへのポインタ	406
システムの動作状態 (動作中、停止中、中断等)	408
その他管理情報	410
システム種別	412

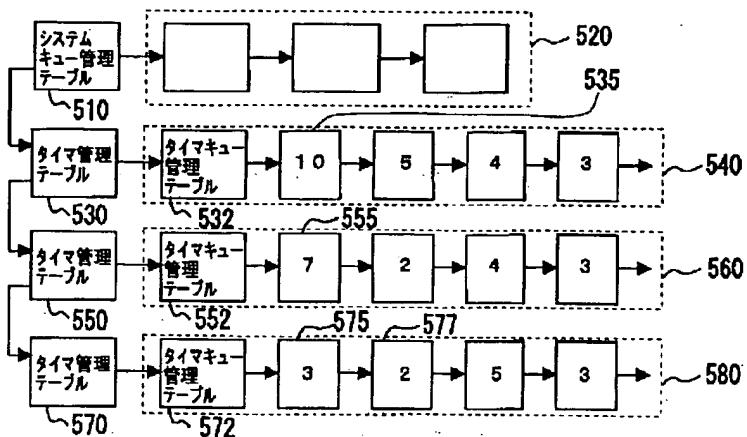
400: システムキュー管理テーブル

【図15】



1500: タイマキュー要素A  
1510: タイマキュー要素B

【図5】



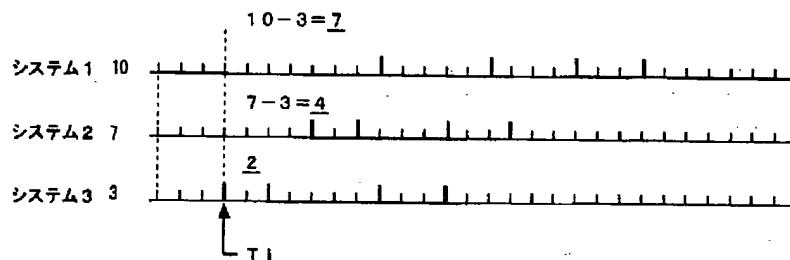
520: 空きタイマキュー  
 530、550、570: タイマ管理テーブル  
 535、555、575、577: タイマキュー要素 (数値は差分タイマ値)  
 540、560、580: タイマキュー

【図24】

	2400	2410	2420
1	10	WAIT	
2	14	WAIT	
3	0	RUN	
4	10	WAIT	
N-1			
N	0	-	
	7	WAIT	

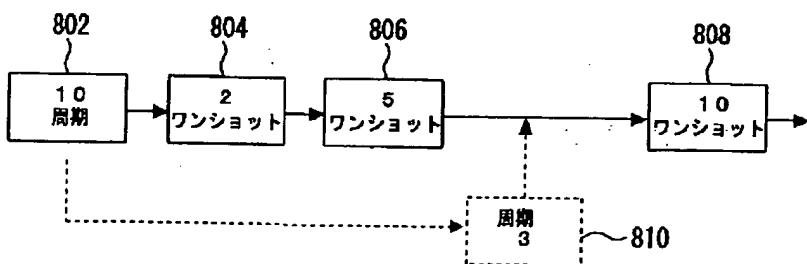
2400: タイマ使用者 (タスク) の識別情報  
 (タスク ID、周期起動時間、etc.)  
 2410: タイマ値  
 2420: タスクの動作状態  
 2450: タイマテーブル

【図6】

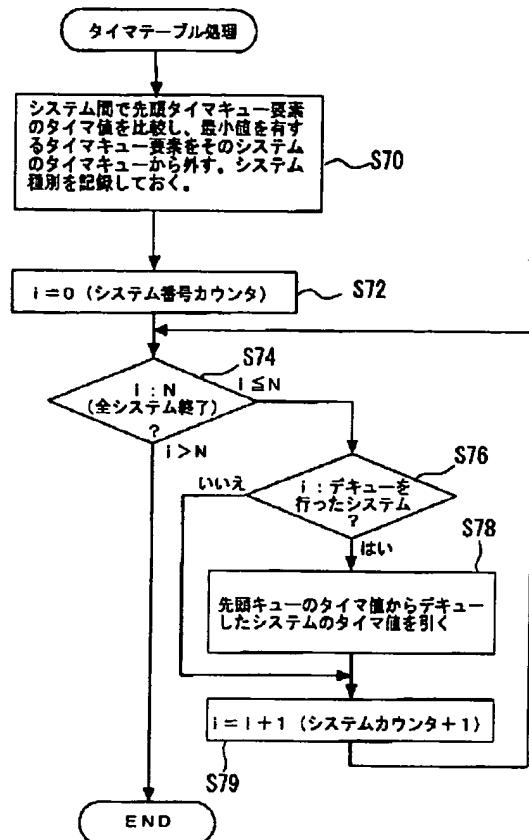


T1: タイマ割込み発生／タイマ再設定 (各キューに7, 4, 2が再設定される)

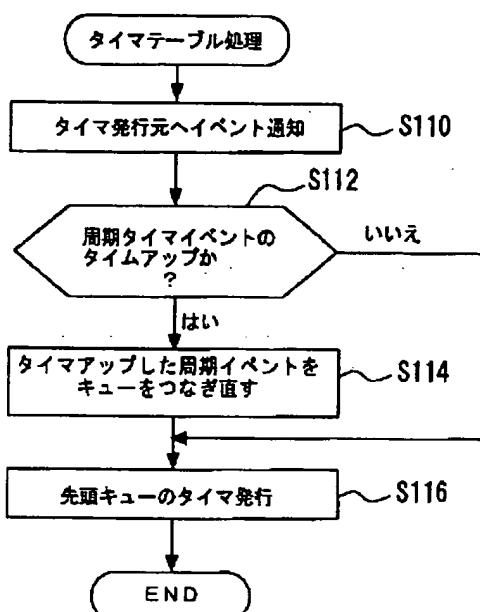
【図8】



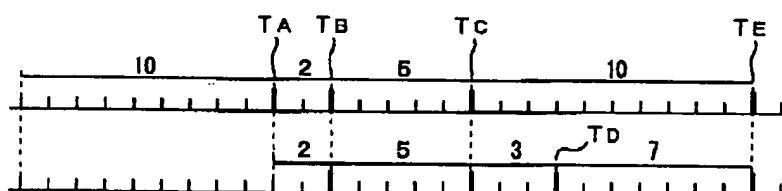
【図7】



【図11】

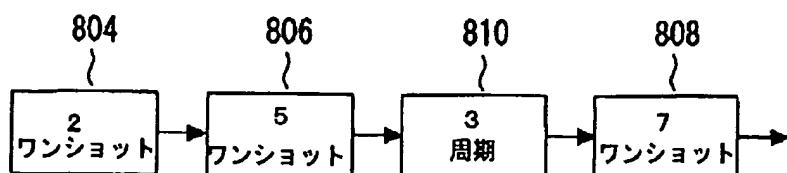


【図9】

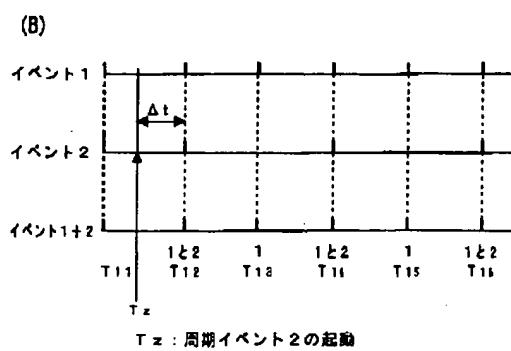
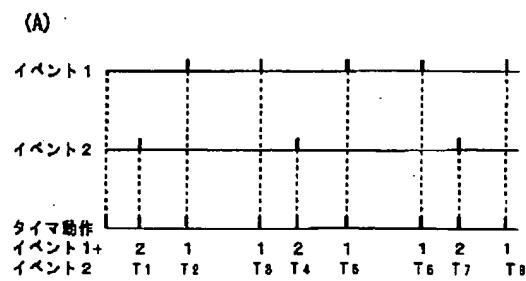


TA: 周期タイマキュー要素802のタイムアップ時刻  
 TB: 周期タイマキュー要素804のタイムアップ時刻  
 TC: 周期タイマキュー要素806のタイムアップ時刻  
 TD: 再設定後の周期タイマキュー要素810のタイムアップ時刻  
 (10=2+5+3)  
 TE: 周期タイマキュー要素808のタイムアップ時刻

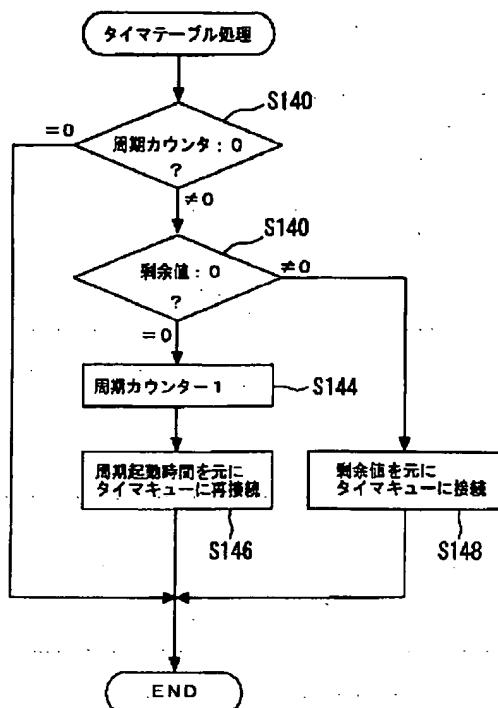
【図10】



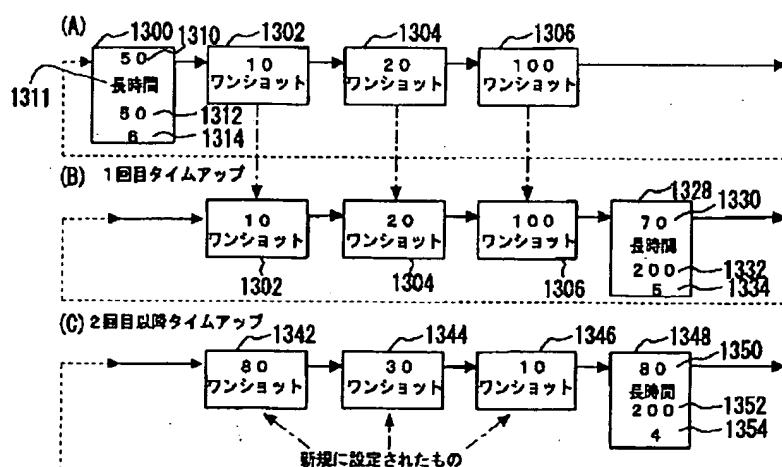
【図12】



【図14】

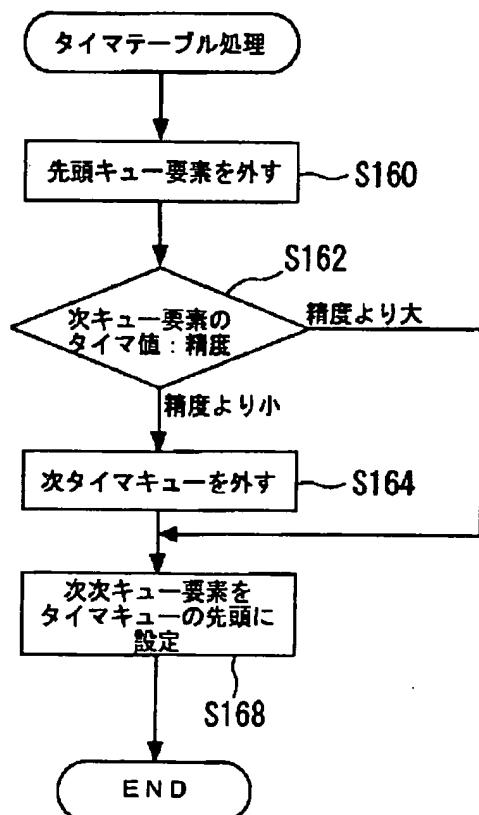


【図13】

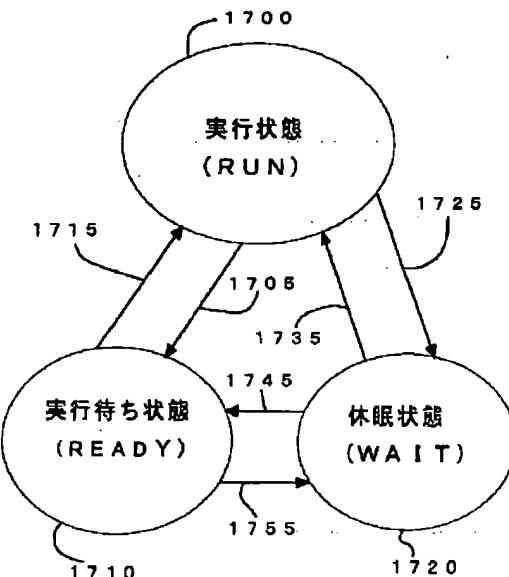


周期: 200、長時間タイマ:  $200 \times 5 + 50 = 1050\text{ms}$   
 1310:  $(50 - 0 = 50)$   
 1312, 1332, 1352: 周期起動時間  
 1314, 1334, 1354: 周期カウンタ  
 1330:  $(200 - 10 - 20 - 100 = 70)$   
 1350:  $(200 - 80 - 30 - 10 = 60)$

【図16】



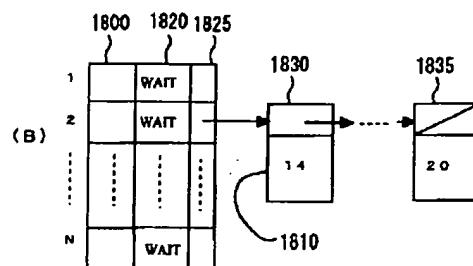
【図17】



【図18】

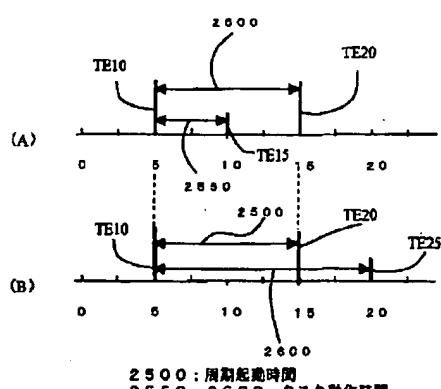
	1850		
	1800	1810	1820
1		10	WAIT
2	...	14	WAIT
3	...	0	RUN
4	...	20	WAIT
⋮	⋮	⋮	⋮
N-1	⋮	0	—
N	⋮	7	WAIT

1800: タイマ使用者(タスク)の識別情報(タスクID,  
実行優先順位, スタックポインタ, e.t.c.)  
 1810: タイマ値  
 1820: タスクの動作状態  
 1850: タイマテーブル

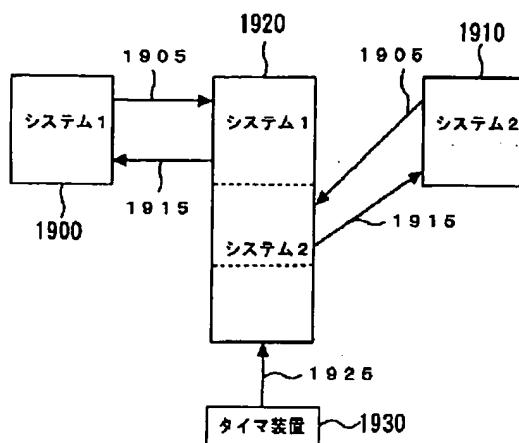


1825: ポインタ  
 1830, 1835: セル

【図25】

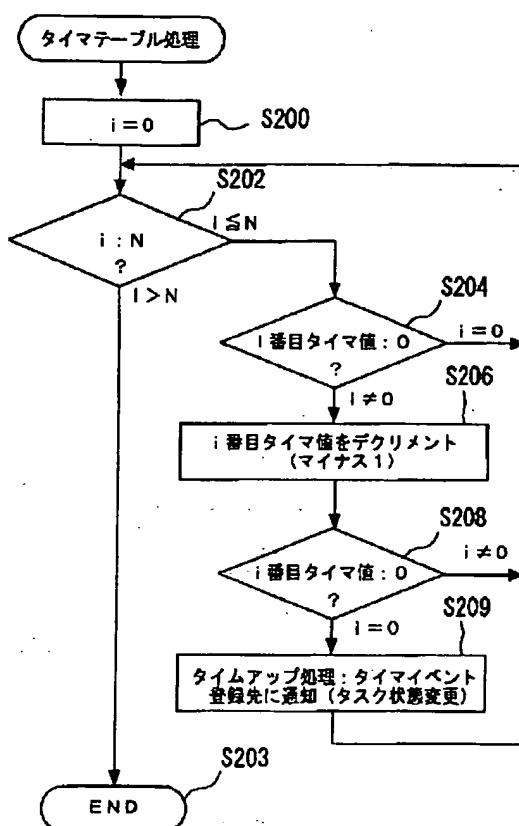


【図19】

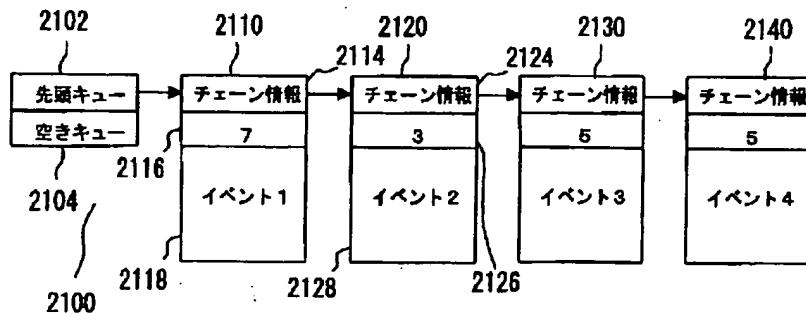


1905 : タイマイベント登録  
 1915 : タイムアップ  
 1920 : タイマ管理テーブル  
 1925 : タイマ割込み

【図20】

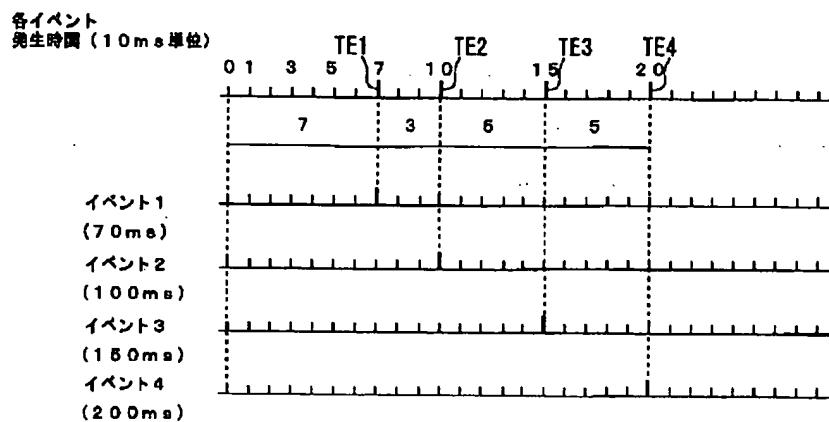


【図21】



2100 : タイマキュー  
 2116, 2126 : タイマ値  
 2110, 2130 : システム1のセル  
 2120, 2140 : システム2のセル

【図22】



【図23】

